Title
Approximation and Control of Skill Based Parallel Service Systems with Homogeneous Service

Author
Grosbard, Dean

Publication Date
2019

Peer reviewed|Thesis/dissertation

Approximation and Control of Skill Based Parallel Service Systems with Homogeneous Service

by

Dean Israel Grosbard

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Industrial Engineering & Operations Research

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Robert C. Leachman, Chair
Professor Rhonda Righter
Professor Jean Walrand

Fall 2019

Approximation and Control of Skill Based Parallel Service Systems with Homogeneous Service

Abstract

Approximation and Control of Skill Based Parallel Service Systems with Homogeneous Service

by

Dean Israel Grosbard

Doctor of Philosophy in Engineering - Industrial Engineering & Operations Research

University of California, Berkeley

Professor Robert C. Leachman, Chair

A skill base parallel service system is comprised of a set of customers of different classes that arrive randomly for service, a set of servers that serve those customers and a set of qualifications that defines which customer classes can be served by which server. Systems of this kind appear in a wide range of applications from the assignment of jobs to employees with different skills to network traffic routing. Literature regarding these systems has almost exclusively been focused on the asymptotic heavy traffic regime. The reason being that such an asymptotic regime is convenient to analyze and allows the derivation of exact results. However, although many applications can be well approximated by an asymptotic regime, many others can not. In this work we are especially concerned with large scale sparse systems where, despite the system being large of scale, each customer class can only be served by a small subset of the servers. After laying foundations for the model in Chapter 1 and exploring structural properties in Chapter 2 we go on to present the two main contributions of this work. In Chapter 3 we develop a set of approximations that compile to a , first of its kind, approximation scheme of matching rates of skill based parallel service system operating under the *first-come-first-serve* or *longest-queue-first* policies. The accuracy of the approximation is verified with extensive simulation experiments where it is shown to provide matching rate estimates with an absolute error of $3\% - 5\%$ for a wide range of traffic intensities. Later, in Chapter 4 we use insights provided by the new approximation to derive weighted versions of the *first-come-first-serve* or *longest-queue-first* and show, through comprehensive simulation testing, that these weighted polices dramatically reduce the waiting time of customers in congested system compared to the original unweighted versions. Finally, we extend the use of the weighted policies to systems with matching rewards and show that, by appropriate choice of weights, these policies can be used by a controller to efficiently trade-off between the rate of reward accumulation and waiting time experienced by the customers

Service Systems; Queuing Theory; Match Rate Approximation; Dynamic Matching.

Dedicated to Yafati Tzan,

Who is with me every step of the way

Contents

List of Figures

List of Tables

Acknowledgments

I would like to first officially thank Google Inc. for providing us with the computation resources that required to conduct the extensive simulation experiments in this paper.
On a more personal note I would like to thank my advisor Professor Robert Leachman who allowed me to come to UC Berkeley and with much patience and wisdom guided me both in the completion of this work and the development of my career. I would also like to thank Professor Dorit S. Hochbaum who advised in the writing of Chapter 2 of this thesis and to the other members of the UC Berkeley Industrial Engineering and Operations Research department faculty and staff whose doors were always open before me. Also, thanks to my other thesis committee members Professors Rhonda Righter and Jean Walrand who provided very valuable advice and encouragement. Thanks go to my fellow PhD students who endured my incessant ramblings about customers classes, servers and the edges that connect them for their feedback, cooperation and of course friendship. Special thanks to Salar Fattahi, Erik Bertelli, Han Feng, Renyuan Xu, Haoyang Cao, Mark Velednitsky, Yonatan Mintz, Quico Span, Alfonso Lobobs, and Mahbod Olfat who on many occasions took the time to hear my ideas out and help get over theoretical, technical and often mental blocks on the path to completing this thesis. Special thanks also to Elias Castro who introduced me to the world of cloud computing. Finally, this thesis is dedicated first and foremost to my beloved wife who left home and country and sacrificed more and worked harder than anyone to allow me to put forth this work. To my dear parents Tuvia and Aviva who supported me from start to finish, believed in me and encouraged me to go off on this adventure. To my in-laws Didi and Anat who hosted my family every summer and gave me the time needed to complete this work. Last but not least to Ofri, Itamar and Airel my wonderful children who filled my life joy every step of the way.

Chapter 1

Introduction

1.1 Skill Based Parallel Service Systems- Literature Review

The foundations of multi-server queueing systems analysis, starting with the early works of Erlang on the $M/M/c$ queuing system in the early twentieth century, rely on two core assumptions, the first being that all interarrival and service times are exponentially distributed, and the second that all servers and all customers are identical. In the basic M/M/c system an arrival stream of identical customers arrive to the system following a Poisson process at rate λ and are served by a set of c identical servers with each customer service requiring an exponentially distributed time period with an average service rate of μ customers per time unit. Under these basic assumptions, the well-known $M/M/c$ closed-form expressions of the steady state distributions can be obtained.

A skill-based parallel service system (SBPSS) can be regarded as a multi-server queueing system in which the assumptions of identical customers and identical servers are relaxed in three ways. First, the single arrival stream of identical customers is replaced with multiple arrival streams of different customer classes. Second, each customer class may only be served by a qualified subset of servers. Third, the time required for a customer to complete service on a qualified server is an exponential random variable with a mean that depends on both the customer class and the server.

In an SBPSS with homogeneous service, the mean service time is not a specific function of the customer-server pair. Instead, the amount of work required depends only on the customer class, and the rate at which work is rendered depends only on the server. This restriction to homogeneous service allows one to properly define ρ, the system traffic intensity, which is the ratio of the rate of work arriving to the system to the cumulative capacity of the servers to remove work from the system. Systems of this type appear in a wide variety of applications and at various scales. In semiconductor manufacturing it is often the case that a set of functionally identical machines perform a set of different operations spread through the overall production sequence or across various products and yet, due to different

hardware configurations or temporary quality restrictions, not every machine can perform every operation. For example, in the case of lithography tools, [51], [18] [33] describe lot-to-lens dedications and limited mask qualifications that restrict tools from performing all the mask layers in a given process. Another canonical example are large call centers, in which streams of incoming calls require agents with different skill sets such as language or proper training. In [5] the authors discuss the assignment of incoming calls to different agent pools under a set of constraints that guarantee fair workload distribution across the agent pools while in [54], the authors discuss the assignment of calls to agents under various trade offs between performance and fairness criteria. In health care operations, the assignment of patients to hospital wards has been modeled and studied as an SBPSS [6]. Skill based parallel service systems are also ubiquitous in modeling the assignment of mobile device communications to physical base stations, where the model is commonly referred to as the *user association problem*. The possible *device-to-base* assignments are limited by spatial locations and hardware limitations, [35],[52]. Another emerging application of SBPSS arises in cloud computing, where requests for virtual machines or containers must be assigned to physical hardware that meet a given set of requirements,[38],[47], [46], [45]. Finally, the growth of the ride sharing economy with companies such as Lyft and Uber has introduced an SBPSS with passengers and drivers assuming the roles of customers and servers, respectively, and the physical locations, as well as service level requirements, determining the divisions of passengers into customer classes and servers into server types, as well the customer class - server type compatibility graph. In [7] the authors use a simplified queuing system framework to derive optimal pricing statagies for the ride sharing platform. In a related work [8] the authors model the ride sharing platforms as a closed queuing network where the servers model drivers and customer arrivals model passenger requests. Server queues represent drivers aggregated into a spatial partition and each customer service moves a server between the server queue at the passengers origin and the server queue at the passengers destination. Passengers finding an empty server queue are dropped. The authors prove that the portion of dropped customers declines exponentially as the number of servers and customers are scaled simultaneously and that the rate of decay is maximized by a certain weighted-LQF(weighted MaxWeight) policy.

A pertinent fact regarding multi-server queues is that the average waiting time experienced by a customer depends both on the traffic intensity of the system and the number of qualified servers. For example, given an $M/M/n$ queue with service rate μ per server and an arrival rate of $\lambda < n \cdot \mu$, if one simultaneously doubles both the customer arrival rate and the number of servers, the resulting system will maintain the same traffic intensity, but the average delay will be reduced. In an SBPSS, the terms "traffic intensity" and "number of servers" can no longer be applied uniformly to every server and every customer class. Some servers may experience a higher traffic intensity than others, while different customers classes may be compatible with different numbers of servers. Although the terms traffic intensity and number of servers can not be applied uniformly to all customer classes and servers in an SBPSS, it is still an intuitive notion that the greater the access to server capacity that a customer class has, the lower the delay that customers of that class will experience. The

impact of the number of servers on the waiting time experienced by a customer class tends to be ignored in the literature concerning the control of skill-based parallel service systems. The vast majority of literature on control of large-scale service-based parallel-service queueing systems considers a system under one of three asymptotic scaling regimes: The efficiency driven (ED) regime, wherein the traffic intensity, the ratio of the workload implied by the customer assignment rates and the capacity of the servers, asymptotically approaches unity [39], the quality driven (QD) regime, wherein the number of servers approaches infinity while the implied traffic intensity is bounded away from 1, and the quality and efficiency driven (QED) regime [26] [27], where the utilization of servers approaches unity and the number of servers approaches infinity simultaneously. In any of these three asymptotic regimes the actual number of available severs becomes irrelevant to determining the control policy. In the ED and QED regimes the overall traffic intensity asymptotically approaches unity and hence the only valid control policies are those that balance the utilization across all servers, regardless of the server counts. This is because any slight imbalance in the utilization of the servers will render the system unstable and the queue lengths of some customer classes will diverge. The QD and QED regimes both pertain to systems with flexible server pools that are each comprised of multiple identical severs having the same set of qualifications, the number of which is scaled to infinity and hence the actual count of qualified servers does not directly impact the system performance.

Asymptotic regimes such as these have been demonstrated to be useful for analyzing the performance of large-scale call centers and communications networks. These systems tend to be comprised of a small number of server pools each containing multiple identical servers and are designed for either efficiency or quality or both. The use of such asymptotic regimes for modeling these systems is therefore natural.

In contrast, herein we are concerned with the waiting times of different customer classes in a large scale parallel system with a sparse qualification set. Such systems contain many servers, yet most customer classes are only served by a small subset of qualified servers. Despite receiving much less attention in the queuing literature, such a system is very common in practice and often arises when the rendering of service by a server to a customer class is limited by geographical, technological or skill constraints. In this type of system both the asymptotic QD and QED regimes are not relevant as there exist customer classes that have access to only a small set of the servers. An ED regime might seem relevant, but the long-term average waiting times under such regimes diverge for any customer classes with a bounded number of servers. The control of an SBPSS outside the context of the ED, QD and QED regimes has received scant attention in literature, albeit there is a rich line of literature concerned with the design of flexible service systems. A striking example of the impact that even a small degree of flexibility has on the performance of an SBPSS was demonstrated in [50]. The authors consider an SBPSS comprised of n servers with service rate $\mu = 1 - p$ each uniquely serving a single customer class with a Poisson arrival process of rate $\lambda = \rho$, plus a single exponential server with a service rate $n \cdot p$ that can serve all customer classes and is assigned the customer class with the longest queue. They prove that when $n \to \infty$, the waiting time for $p = 0$ scales as $(1 - \rho)^{-1}$ but for any $p > 0$ the waiting

time only scales as $\log_{(1-p)^{-1}}((1-\rho)^{-1})$. In another paper [49] the asame uthors consider an Erdős-Rényi bipartite graph and demonstrate that for a graph with n nodes on each set and an arc probability of $ln(n)/n$, a vanishing delay may be achieved as $n \to \infty$ when customers are first batched for a time period and then assigned to servers by a maximum cardinality matching.

1.2 Basic Notation

In this work we will use $\mathbb{R}_+$ and $\mathbb{R}_{++}$ to describe the set of all non-negative and strictly positive real numbers. The bold font notation is reserved for complete vectors and matrices while a plain font is used for the entries so that $\mathbf{a}$ is a vector while a_i is the $i-$th entry of the vector $\mathbf{a}$. The abbreviation *w.p* is a shorthand for "with a probability measure of 1"

1.3 The Skill Based Parallel Service System Model

A Skill-Based Parallel Service System (SBPSS) with homogeneous service (HS) is described by the following six element tuple $\mathcal{F} = (\mathcal{I} \cup \mathcal{J}, E, \boldsymbol{\lambda}, \mathbf{s}, \boldsymbol{\mu})$. The first three elements are the sets of nodes and edges of the undirected bipartite graph that defines the topology of the system. The set of customer classes are denoted by $\mathcal{I} = \{1, \ldots, m\}$ and a set of servers, $\mathcal{J} = \{1, \ldots, n\}$. For a given server $i \in \mathcal{I}$ and customer class $j \in \mathcal{J}$ we say that "*i is compatible with j* " or equivalently "*i can be served by j* if $(i, j) \in E$ where $E \subseteq \mathcal{I} \times \mathcal{J}$ is the set of all compatible customer-server pairs. For any system, unless otherwise stated the following assumption is made:

Assumption 1.3.1 (Connected Compatibility Graph)**.** *The graph $G = (\mathcal{I} \cup \mathcal{J})$ is connected.*

For a given set of customer classes $I \subseteq \mathcal{I}$ or servers $J \subseteq \mathcal{J}$ we denote their set of compatible servers or customer classes respectively by

$$\begin{aligned} \partial(I) &= \{j \in \mathcal{J} | i \in I, (i, j) \in E\} \\ \partial(J) &= \{i \in \mathcal{I} | j \in J, (i, j) \in E\} \end{aligned} \tag{1.1}$$

and with slight abuse of notation we let $\partial(x) = \partial(\{x\})$. In a general SBPSS the rate at which server j serves customers of class i is a random variable with a mean denoted by μ_{ij}. In this paper we focus on specific type of system which we refer to as an SBPSS with homogeneous service defined as follows:

Definition 1.3.1 (Homogeneous Service)**.** *An SBPSS is said to have homogeneous service if and only if there exist sets $\{\mu_j | j \in \mathcal{J}\}$ and $\{s_i | i \in \mathcal{I}\}$ of strictly positive scalars such that*

$$\mu_{i,j} = \frac{\mu_j}{s_i} \quad \textit{for all } (i, j) \in E \tag{1.2}$$

The values s_i, μ_j in definition 1.3.1 can be thought of as the amount of work required by a customer of class i and the rate of service provided by a server j respectively. Therefore, in a SBPSS with homogeneous service the portion of the overall workload introduced into the system by a customer class i is independent of the service policy and is given by

$$\eta_i = \lambda_i s_i, \quad i \in \mathcal{I} \tag{1.3}$$

where $\lambda_i, i = 1, \ldots, n$ is the arrival rates customer $i = 1, \ldots, n$. A specific case of a SBPSS with homogeneous service is the SBPSS with server dependent service times.

Definition 1.3.2 (Server Dependent Service Times)**.** *A SBPSS has server dependent service times if and only if*

$$\mathbf{s} = s\mathbb{1}_m \text{ for some } s \in \mathbb{R}_+ \tag{1.4}$$

For the remainder of this paper we will consider only SBPSS with homogeneous service unless otherwise stated. For sets of customer classes I and servers J we denote

$$\begin{aligned} \lambda_I = \sum_{i \in I} \lambda_i \quad , \quad \eta_I = \sum_{i \in I} \eta_i, \qquad & \forall I \subseteq \mathcal{I} \\ \mu_J = \sum_{j \in J} \mu_j, \qquad & \forall J \subseteq \mathcal{J} \end{aligned} \tag{1.5}$$

The matching rates in a SBPSS depend both on the systems structure as described above and the matching policy used to determine which customer-server matches are made in real time. In this paper we restrict discussion to non-preemptive, non-idling, *head-of-line*(HOL) matching policies. A system operating under a non idling, non preemptive matching HOL policy behaves as follows:

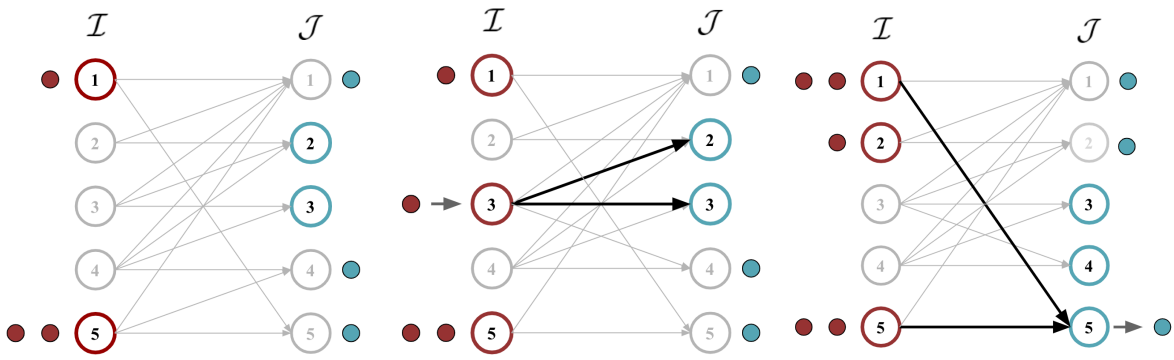
1. The queue of class i customers is said to be *active* if there are customers waiting in the queue and *inactive* otherwise

2. A server j is considered *available* if it is idle and *unavailable* otherwise

3. Upon arrival, a customer of class i finding a set of available qualified servers will immediately match with an available server and will be served until completion or, if no qualified idle servers exist, wait in a i class customer queue.

4. Upon completion of service a server j finding a set of active qualified customer class queues will immediately be matched with a customer and will be served until completion or, if no qualified active queues exist, go idle and become available.

5. At every match instance the customer matched is the longest waiting customer of his respective customer class



(a) No arc between an active queue and idle server.

(b) customer departure → server matching policy

(c) ustomer arrival → customer matching policy

Figure 1.1: An SBPSS under a non-idling, non preemptive, head-of-the-line policy

Note that under a non-preemptive and non-idling policy assignment decisions need only be made at the instance of either an arrival of a customer or the departure of a customer, which coincides with a completion of service. Furthermore, we assume both service and interarrival distributions are continuous and hence at such times, *w.p.* 1, there will either be a single active queue with multiple qualified idle servers or a single idle server with multiple qualified active queues. The matching policy can therefore be regarded as comprised two separate policies:

Customer Matching Policy: A customer of class i arrives. If a set of compatible idle severs $J \subseteq \partial(i)$ exists, the customer matching policy $\Psi_{\mathcal{I}}$ determines which idle server the customer is assigned to.

Server Matching Policy: A server of type j completes an assignment. If a set of active compatible customers queues exist, the server assignment policy $\Psi_{\mathcal{I}}$ determines which waiting customer is assigned to the server.

A pair of customer and server assignment rules constitute a control policy $\Psi = (\Psi_{\mathcal{I}}, \Psi_{\mathcal{J}})$. In order to properly describe a matching policy Ψ we must first introduce our object of interest which is the stochastic process $\zeta_{\mathcal{F},\Psi}(t) = (A, S, D, Z, X, Q, W, I)$ the components of which we will now describe. Let $\hat{A} = \{\hat{A}_i | i \in \mathcal{I}\}$ be a set of independent renewal counting processes $\hat{A}_i = \{\hat{A}_i(t), t \geq 0\}$ with interarrival-time distribution having mean 1 and finite variance. The arrival process for class i is the time-scaled renewal process $A_i = \{A_i(t) = \hat{A}_i(\lambda_i t), t \geq 0\}$ and $A = \{A_i | i \in \mathcal{I}\}$ is the set of customer class arrival process. The service duration for a customer of class i being served by a server $j \in \partial(i)$ is a positive valued random variable with mean μ_{ij}^{-1}. Let $\hat{S} = \{\hat{S}_{ij} | (i,j) \in E\}$ be a set of independent renewal counting processes $\hat{S}_{ij} = \{\hat{S}_{ij}(t), t \geq 0\}$ with an identical interarrival-time distribution having mean 1 and finite variance. The service process of a pair $(i,j) \in E$ is the time scaled renewal

process $S_{ij} = \{S_{ij}(t) = \hat{S}_{ij}(\mu_{ij}t), t \geq\}$ so that $S_{ij}(t)$ counts the number of i class customer assignments server j has completed after spending t time units serving customers of class i and $S = \{\hat{S}_{ij} | (i,j) \in E\}$ is the set of qualified customer class-server service processes. For any $(i,j) \in E$ let

$$Z_{ij}(t) = \begin{cases} 1 & , \textit{ if server } j \textit{ is serving a customer of class } i \textit{ at time } t \\ 0 & , \textit{ otherwise} \end{cases} \tag{1.6}$$

The number of class i customers that have completed service on server $j \in \partial(i)$ by time $t \geq 0$ is thus given by:

$$D_{ij}(t) = S_{ij}\left(\mu_{ij} \int_0^t Z_{ij}(\tau) d\tau\right), \quad \textit{for all } (i,j) \in E \tag{1.7}$$

and $D = \{D_{ij}(t), (i,j) \in E\}$ is the set of customer class-server specific departure processes. The count of class i customers in the system at time t, is denoted by $X_i(t), i \in \mathcal{I}$ where

$$X_i(t) = A_i(t) - \sum_{j \in \partial(i)} D_{ij}(t), \quad i \in \mathcal{I} \tag{1.8}$$

while the number of class i customer in the queue at time t is given by

$$Q_i(t) = X_i(t) - \sum_{j \in \partial(i)} Z_{ij}(t) \tag{1.9}$$

with $X = \{X_i(t), \in \mathcal{I}\}$ and $Q = \{Q_i(t), \in \mathcal{I}\}$. Due to the assumption of a HOL policy the waiting time of the longest waiting customer in the queue of class i customers at time t is given by:

$$W_i(t) = \begin{cases} t - \min\{\tau | A_i(\tau) > \sum_{j \in \partial(i)} D_{ij}(t) + Z_{ij}(t)\} & , \textit{ if } Q_i(t) > 0 \\ 0 & , \textit{ otherwise} \end{cases} \tag{1.10}$$

The idleness of a server j at time r is the elapsed duration between the last instance when server j was busy and the current time t and is given by

$$I_j(t) = t - \sup\{\tau | \sum_{i \in \partial(j)} Z_{ij}(\tau) > 0, \tau \leq t\} \tag{1.11}$$

Having defined the stochastic process $\zeta_{\mathcal{F},\psi}$ we restrict the discussion to matching policies that are Markovian with respect to $\zeta_{\mathcal{F},\psi}$. A pair of Markovian matching policies $\Psi = (\Psi_{\mathcal{I}}, \Psi_{\mathcal{J}})$ can now be described by associating with each policy functions $\psi_{\mathcal{I}}, \psi_{\mathcal{J}} : \mathbb{N}^m \times \{0,1\}^{m \times n} \times \mathbb{R}_+^m \times \mathbb{R}_+^n \to \mathbb{N}^m \times \{0,1\}^{m \times n}$ such that:

$$\begin{array}{ll} \psi_{\mathcal{I}}(Q(a_k^-), Z(a_k^-), W(a_k^-), I(a_k^-)) & = (Q(a_k), Z(a_k)) \\ \psi_{\mathcal{J}}(Q(d_k^-), Z(d_k^-), W(d_k^-), I(d_k^-)) & = (Q(d_k), Z(d_k)) \end{array} \tag{1.12}$$

where $a_k, d_k k \in \mathbb{N}$ are the instances of customer arrival and service departures given by:

$$\begin{array}{lll} a_k = \inf\{t | \sum_{i\in\mathcal{I}} A_i(t) \geq k\} & , & a_k^- = \sup\{t | \sum_{i\in\mathcal{I}} A_i(t) < k\} \\ d_k = \inf\{t | \sum_{(i,j)\in E} D_{ij}(t) \geq k\} & , & d_k^- = \sup\{t | \sum_{(i,j)\in E} D_{ij}(t) < k\}. \end{array} \tag{1.13}$$

The performance measures of interest for the SBPSS can now be rigorously defined. The system matching rates are given by the limit

$$r_{ij} = \lim_{t\to\infty} \frac{D_{ij}(t)}{t}, \quad \textit{for all } (i,j) \in E \tag{1.14}$$

and we define the utilization of a server $j \in \mathcal{J}$ by

$$\rho_j = \lim_{t\to\infty} \frac{1}{t} \sum_{i\in\partial(j)} Z_{ij}(t) \tag{1.15}$$

The long term avg. waiting time and queue length of a customer class $i \in \mathcal{I}$ are defined as

$$Wq_i = \lim_{t\to\infty} \frac{1}{A(t)} \int_0^t Q_i(\tau)d\tau \quad \text{and} \quad Lq_i = \lim_{t\to\infty} \frac{1}{t} \int_0^t Q_i(\tau)d\tau \tag{1.16}$$

respectively. The existence of the limits in (1.14),(1.15), (1.16) depends both on the structure $\mathcal{F}$ and the matching policies Ψ. A method for determining the existence of finite limits for an arbitrary non-idling, non-preemptive policy is not known. However, in the remainder of this work we will focus on two server matching policies *First-Come-First-Served*(FCFS) and *Long-Queue-First* (LQF) and a single customer matching policy *Assign-Longest-Idle-Server* (ALIS) for which the existence of the limits (1.14),(1.15), (1.16) can be proven under certain conditions to be discussed in Chapter 3. The three aforementioned polices can be formally defined as:

- **FCFS:** *Server j will match with the longest waiting customer from queue i' where*

$$i' = \text{argmax}\{W_i(t) | i \in \partial(j)\}$$

- **LQF:** *Server j will match with the longest waiting customer from queue i' where*

$$i' = \text{argmax}\{Q_i(t) | i \in \partial(j)$$

- **ALIS:** *Customer of class i will match with server j' where*

$$j' = \text{argmax}\{I_j(t) | j \in \partial(i)\}$$

In Chapter 4 we will also introduce weighted versions of these policies.

1.4 Waiting Time in Skill Based Parallel Service Systems

A $M/M/n$ multi-server queue can be considered as a special case on an SBPSS, where $E = \mathcal{J} \times \mathcal{I}$ and the service times are homogeneous and exponentially distributed. In an $M/M/c$ queue, under any non-idling service policy that does not discriminate between servers (i.e, the assignment decision does not require knowledge of the server identity) the server utilization must be uniform and will be given by

$$\rho = \min\{1, \frac{\sum_{i\in\mathcal{I}} \eta_i}{\sum_{j\in\mathcal{J}} \mu_j}\} \tag{1.17}$$

and hence $\rho_j = \rho$ for all $j \in \mathcal{J}$. Even if different customer classes exist (a $\Sigma M/M/n$ queue), the utilization of the servers directly determines the average waiting-time of across all customer classes and is given by

$$W_i = \left[\left(n\mu - \sum_{i\in\mathcal{I}} \lambda_i\right) \cdot \left(1 + (1-\rho) \cdot \left(\frac{n!}{n \cdot \rho^n}\right) \cdot \left(\sum_{k=0}^{n} \frac{(n\rho)^k}{k!}\right)\right)\right]^{-1} \tag{1.18}$$

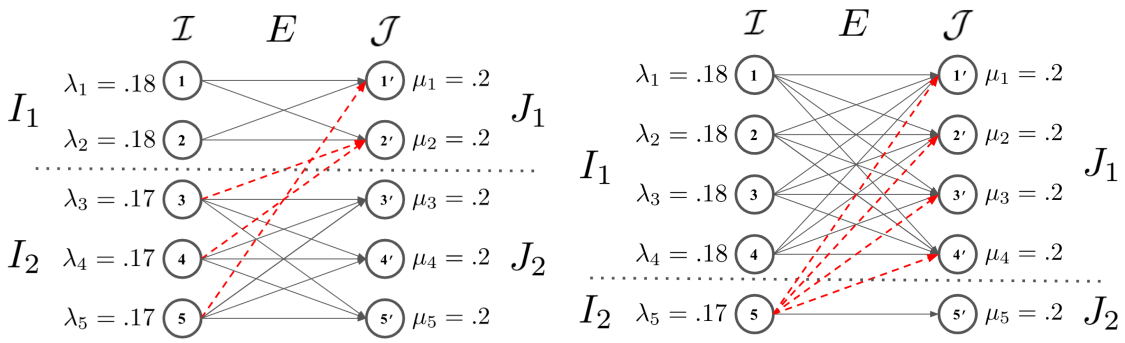
Furthermore, under a FCFS-ALIS or LQF-ALIS policy, by mere symmetry, the workload of each customer class will be equally distributed across the servers and hence

$$r_{ij} = \frac{\lambda_i}{n} \tag{1.19}$$

Clearly, for an SBPSS where $E \subsetneq \mathcal{I} \times \mathcal{J}$ one may expect a higher waiting time compared to a multi-server queue with full qualifications. To provide some intuition as to why a higher waiting time may be expected we use the Sakasegawa approximation [42] of the waiting time in a system with n identical servers, exponential service and interarrival times. The closed form expression in (1.18) is exact, however the simplicity of the approximation enables us to gain more insight. In an $M/M/n$ queue, recalling that $\rho_j = \rho$ for all $j \in \mathcal{J}$ the avg. waiting time can be well approximated by:

$$\tilde{W}q(\rho, n, \mu) \approx \frac{\rho^{\sqrt{2(n+1)}-1}}{n(1-\rho)} \cdot \frac{1}{\mu} \tag{1.20}$$

From the approximation of (1.20) it is clear that for a fixed service rate of μ the waiting time is increasing in ρ and decreasing in n. In an $M/M/n$ queue each customer class has access to n servers each utilized at the a rate of ρ but in a general SBPSS this may not be the case. Let us consider the example of the SBPSS depicted in Figure 1.2a. The customer set $I_1 = \{1, 2\}$ may only be served by servers in server set $J_1 = \{1', 2'\}$ and the servers in the set $J_2 = \{3', 4', 5'\}$ can only serve customer classes in the set $I_2 = \{3, 4, 5\}$. However, customer classes in I_2 can be served by servers in J_1 as indicated by the dotted arcs in Figure1.2a.



(a) Over Utilized Subsystem

(b) Under Sized Subsystem

Figure 1.2: Two types of sub systems in a flexible service system

The service of I_1 customer classes alone requires that the avg. utilization of the servers in J_1 be at least $.9 = (2 \cdot 0.18)/(2 \cdot 0.2)$ while the utilization of servers in J_2 can be no higher than $.85 = (3 \cdot 0.17)/(3 \cdot 0.2)$ which can be only achieved by servicing all customers of the classes in I_2. The avg. utilization of the servers in this system is $.87 = (0.18 \cdot 2 + 0.17 \cdot 3)/(5 \cdot 0.2)$ which is lower than the minimal avg. utilization of servers in J_1 and higher than the maximum avg utilization of servers in J_2. In this specific case we also have both $I_1 \times J_1 \subseteq E$ and $I_2 \times J_2 \subseteq E$ and so we can obtain both a lower bound on the waiting time of customers classes in set I_1, $Wq_{I_1} \geq \tilde{W}q(.9, 2, .2) \approx 21.46$ and an upper bound on the waiting time of customers classes in set I_2, $Wq_{I_2} \geq \tilde{W}q(.85, 3, .2) \approx 8.25$. These are upper and lower bounds because the arcs in $E \cap I_2 \times J_1$ can increase the avg. utilization of the servers in J_1 beyond .9 while reducing the avg. utilization of servers in J_2 below .85 and will inevitably do so under any non-idling service policy. An idling service policy that avoids assignment on the arcs in $E \cap I_2 \times J_1$ and is otherwise non idling will indeed achieve the lower and upper bounds and result in an average waiting time of 13.72 time unites. This does not necessarily imply that the avg. waiting time in the system is minimized. The fact that $5.45 = \tilde{W}q(.87, 5) < \tilde{W}q(.85, 3, .2) < \tilde{W}q(.9, 2, .2)$ suggests that the arcs in $E \cap I_2 \times J_1$ may be utilized in a manner that, despite increasing Wq_{I_1} beyond the lower bound waiting time of $\tilde{W}q(.85, 2)$, will reduce the overall avg. waiting time in the system. However results of a simulation experiment indicate that under a FIFO-ALIS policy the average waiting time in the system is 14.3 time units, greater than the waiting time when the system is decomposed.

The set $I_1 \cup J_1$ of the SBPSS in Figure 1.2a is an example of an over-utilized subsystem: a sub-system for which, under any feasible assignment matrix, the average utilization of the servers will be higher than the average system utilization ρ and the customer classes in the subsystem can not be served by any other servers outside the subsystem. The average utilization of an SBPSS under any feasible assignment is fixed at ρ, hence if an over-utilized subsystem exists there must also exist within the same SBPSS an under utilized subsystem,

that is, a sub-system such that under any feasible assignment the average utilization of the servers is lower than ρ and the servers in the system can not serve any customer class outside the subsystem. In the SBPSS in Figure 1.2a the system $I_1 \cup J_1$ is an over-utilized sub system and $I_2 \cup J_2$ is an under-utilized subsystem. Both sub-systems have a complete qualification set and hence we can definitively state that under any non-idling policy, or even a policy that prevents assignments on $E \cap I_2 \cup J_1$, we will have $Wq_{I_1} > Wq_{I_2}$. However is is not always the case that the waiting time of an under-utilized subsystem is lower than that of an over-utilized subsystem. Let us consider the example in Figure 1.2b. The sets $I_1 = \{1, 2, 3, 4\}$ and $J_1 = \{1', 2', 3', 4'\}$ form an over-utilized subsystem. The customers in I_1 can only be served by the servers in J_1 and therefore under any feasible assignment the utilization of the servers in J_1 can be no lower than $(0.18 \cdot 4)/(0.2 \cdot 4) = 0.9$ while the overall utilization of the system is $\rho = (0.18 \cdot 4 + .17)/(0.2 \cdot 5) = 0.89$. The sets $I_2 = \{5\}$ and $J_2 = \{5'\}$ form an under-utilized sub-system as server $5'$ can only serve customer class 5 and by serving all class 5 customers will achieve a utilization of $\rho_5 = .85 = .17/.2 < .89$. However, despite subsystem $I_1 \cup J_1$ being over-utilized and $J_2 \cup I_2$ being under utilized it is not necessarily true that $Wq_{I_1} > Wq_{I_2}$. Since both sub-systems have complete qualification sets, if we were to isolate both subsystems by avoiding all assignments on the arcs in $I_2 \times J_1 \cap E$ we will have $Wq_{I_1} = \tilde{W}q(.9, 4) = 1.99 < Wq_{I_2} = \tilde{W}q(.85, 1) = 5.67$ and the average waiting time in the system will be $Wq = 4 \cdot 0.18 \times Wq_1 + 0.17 \cdot Wq_2 = 2.396 < Wq_2$. The customers in the under-utilized subsystem $I_2 \cup J_2$, if isolated, will experience a longer waiting time than the system average because despite the low utilization of the J_2 servers the subsystem has only a single server and is thus an under-sized subsystem. Not only is the waiting time of customers in the under-utilized subsystem higher in isolation compared to the over-utilized systems but it is also possible to improve the overall average waiting in the system by assigning workload from the under-utilized system to the over-utilized system. Let us split customer class 5 such that with probability $q = \frac{4}{5} \cdot \frac{1}{17}$ an arriving customer of class 5 can only be served by the servers of J_1 and with probability $1 - q$ it can only be served by servers of J_2. The resulting utilization of J_1 servers is .91 and of the utilization of the servers in J_2 will .81. If we now apply a FIFO-ALIS service policy, we will have $Wq_{I_1} = \tilde{W}q(.91, 4) = 2.265, Wq_{I_2} = q \cdot Wq_{I_1} + (1-q) \cdot \tilde{W}q(.81, 1) = 4.17$ and the resulting average waiting time in the system will be $Wq = (4 \cdot 0.18 + q \cdot 0.17) \cdot \tilde{W}q(.91, 4) + (1-q) \cdot 0.17 \cdot \tilde{W}q(.81, 1) = 2.34$, which is lower than the average waiting time in case we isolate both systems. The system in Figure 1.2b is a counter example to a common misconception that improved load balancing means improved system performance. The two systems in Figure 1.2 demonstrate that the average waiting time experienced by a customer class in an SBPSS is not solely dependent on the number of servers in the system and average system utilization, and that balancing the system loading does not necessarily improve the overall system performance. Even accounting for the number of servers available to a specific customer class and the utilization of those servers is not enough information to predict the average waiting time of the customer class, as we can see in the following example. Let us consider the following series of systems indexed by $n \in \mathbb{N}$, as illustrated in Figure 1.3 for $n = 4$. The system which we will refer to as the $M/M/n - \textit{plus 1}$ is constituted of a customer class set $\mathcal{I}_n = I_0 \cup I_n$ where $I_0 = \{0\}$

and $I_n = \{1, \ldots, n\}$, server set $\mathcal{J} = J_0 \cup J_n$ where $J_0 = \{0\}$ and $J_n = \{1', \ldots, n'\}$ and a qualification set $E = E_0 \cup E_n$ where $E_0 = \{(0,0), (0,1)\}$ and $E_n = I_n \times J_n$. Let the arrival rates of all customers be identical, $\lambda_i = \eta/n, i \in \mathcal{I}$ and similarly let the service rates be equal across all servers with $\mu_j = 1/n, j \in \mathcal{J}$. Note that regardless of the size of n, customer class 0 has only 2 servers available to it; $\partial(0) = \{0', 1'\}$ and that if utilization is balanced across the servers, both servers, and all other servers in the system, have an equal utilization of ρ. The neighbourhood of customer class 0 does not change with n, however as illustrated

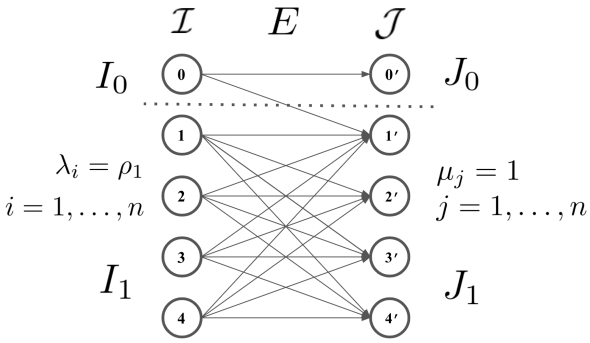


Figure 1.3: An $M/M/n - plus\ 1$ system with $n = 4$

in Figure 1.4 for the case of $\eta = .99, .95, .9$. the average waiting time of class 0 decreases rapidly as n grows large under the FCFS-ALIS policy. This decline in waiting time can be explained by the increases in the portion of class 0 customers that are assigned to server 1 as n increases. As the system size increases the average waiting time and queue lengths of customers in classes 1 to n decrease. These complex dynamics that arise in the rather simple

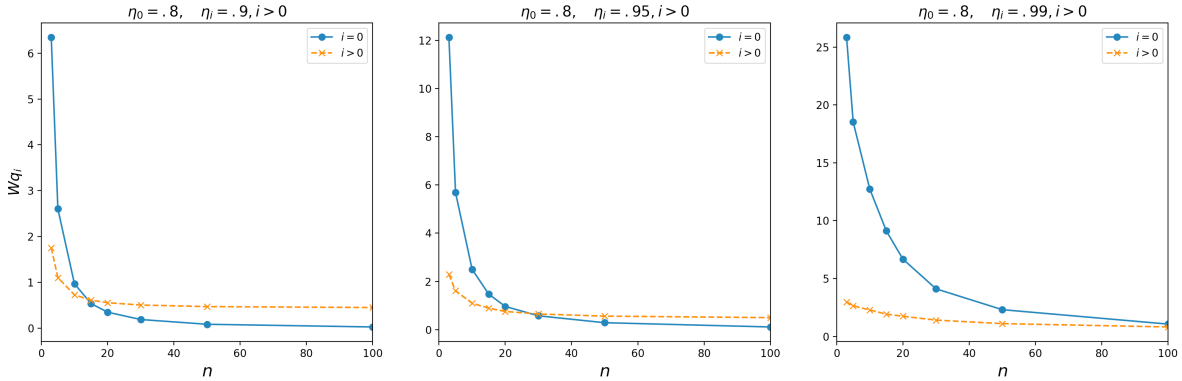


Figure 1.4: Waiting time 0, $i > 0$ customer classes in the $M/M/n - plus\ 1$ system

examples of this section provide motivation for . The remainder of the thesis is arranged as

follows: In the remaining section 1.5 of Chapter 1 we will describe several canonical system structure to be used repeatedly in subsequent chapters. Chapter 2 of the thesis we focus structural properties of the underlying bipartite graph of the SBPSS, mainly the min-max-fair decomposition of the graph which we define and use to state the conditions under which the SBPSS can be stabilized by a non-idling policy. The main contributions of this thesis appear in Chapters 3 and 4. In Chapter 3 we first re-derive the approximation of [14] for the infinite FCFS bipartite matching sequence of [3] as a maximum entropy approximation and demonstrate the advantage of the approximation over others in literature using both numerical and simulation experiments. In section 3.2 we define the ALIS infinite bipartite matching and provide a novel fluid dynamics based approximation of the models matching rates. The FCFS and ALIS infinite bipartite matching sequence approximations of Sections 3.1 and 3.2 are combined in Section 3.3 to produce the key contribution of this work which is, to the best of our knowledge, the first approximation scheme for the matching rates for a subcritical SBPSS with homogeneous service under the FCFS-ALIS and LQF-ALIS service policies. The accuracy of the approximations scheme is demonstrated with extensive simulation experiments for a wide range of cases. In Chapter 4 we begin by leveraging upon the approximations derived in Chapter 3 to expose inherent flaws in the FCFS-ALIS and LQF-ALIS policies. Having exposed these flaws in section 4.4 we propose new weighted versions of the FCFS-ALIS and LQF-ALIS policies and demonstrate their efficacy with simulations. In Section 4.6 we extend the weighted policies to accommodate systems with matching rewards and show, using simulation experiments, that such weighted policies can be used to effectively trade-off the customer waiting time and matching reward rate. Finally, in Chapter 5 we conclude with a summary of the contributions of the thesis and review of existing gaps in the work followed by recommendations for possible future research directions.

1.5 Chains, Grids, Maps and Erdős-Rényi Graphs

The main contributions in this paper are in the form of approximations in Chapter 3 and control heuristics in Chapter 4. That being the case, we find it necessary to investigate results from applications over a wide range of SBPSS types and at various scales. The compatibility graphs we are interested in are ones that have the following three properties: First, they should be scalable so that we may observe both small instances that can be solved analytically and compared to our heuristics as well as large scale instances for which the analytic methods are not applicable and the efficacy of our heuristics can be demonstrated. The second desired property is sparsity; in general, for a large SBPSS with a dense graph, the behaviour of the SBPSS tends to resemble that of a complete system. Hence we focus on sparse graphs where the approximation and control of the system is not trivial. Finally, we wish to focus on feasible cases where the SBPSS has the capacity to process the arriving workload. In order to meet this final requirement we need to use certain parameters to generate the graphs as will be discussed.

Homogeneous bipartite Erdős-Rényi

In a Homogeneous Erdős-Rényi type SBPSS there is an equal number $n = m$ of customer classes and servers and the set of edges $E \subseteq \mathcal{I} \times \mathcal{J}$ is generated by drawing a random sample from a random variable $X \sim Bernoulli(p)$ for every pair $(i, j) \in \mathcal{I} \times \mathcal{J}$ and including the (i, j) in the set of edges if the sample equals 1 and excluding it if the sample is 0 so that every pair of nodes $i \in \mathcal{I}$ and $j \in \mathcal{J}$ are connected with a probability p. Let $\omega = \omega(n)$ be some function of n. We may, without loss of generality, rewrite p as

$$p = \frac{log(n) + \omega(n)}{n} \tag{1.21}$$

The probability that a random bipartite graph $ER(n, p)$ with edge probability p as $n \to \infty$ contains a perfect matching is given by(see Theorem 6.1 of [22]):

$$\lim_{n\to\infty} \mathbb{P}\left(ER(n,p) \text{ has a perfect matching }\right) = \begin{cases} 0 & \text{if } \omega \to -\infty \\ e^{-2e^{-c}} & \text{if } \omega \to c \\ 1 & \text{if } \omega \to \infty \end{cases} \tag{1.22}$$

Hence we let

$$p = \frac{2\log(n)}{n} \tag{1.23}$$

This specific probability is chosen as we wish on the one hand to generate a sparse graph such that $w.p.$ 1 the graph density $n^{-2}\mathbb{E}_X|E| \to 0$ exponentially fast as $n \to \infty$ while on the other hand we have a high probability that for a random choice of workload arrival and service rate unit sum vectors $\boldsymbol{\eta}, \boldsymbol{\mu}$, the graph would be feasible by Lemma 3.2 of [49].

k-Torus

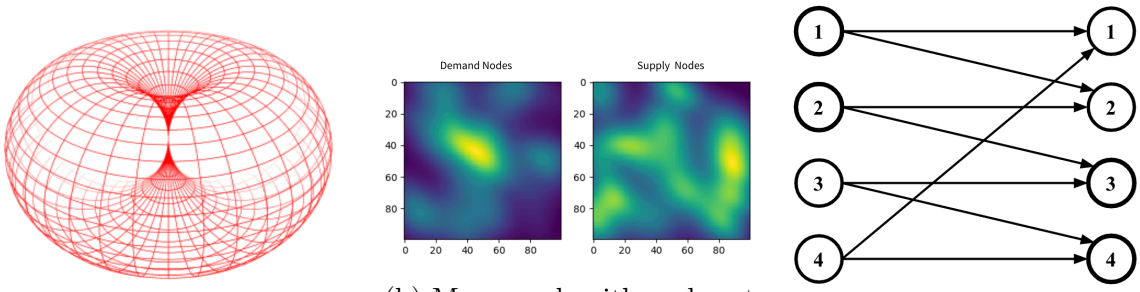
The graph of a k torus SBPSS of size n denoted by $Torus(n \times n, k)$ has n^2 customer classes $\mathcal{I} = \{(i_1, i_2)|i = 1 \ldots, n\}$ and n^2 servers $\mathcal{J} = \{(j_1', j_2')|j = 1, \ldots, n\}$ that are aligned on a tours grid such that each pair of customer and server nodes is adjacent if $d_n(i_1, j_1') + d_n(i_2, j_2') \leq$ where

$$d_n(i, j) = \min(mod_n(i - j), mod_n(j - i)) \tag{1.24}$$

Hence, the topology of the graph is completely symmetrical and the nodes are indistinguishable.

Map

The graphs of a Map SBPSS are intended to simulate a geographical distribution resources, for example, the case of ride-sharing where the passenger requests and driver supply tend to concentrate in specific regions. For this purpose we generate both customer workload and service rate surfaces using random mixture of Gaussian distributions. For a grid map



(a) Torus grid

(b) Map graph with nodes at grid intersections

(c) Homogeneous 2-chain

Figure 1.5: Illustrations of Different Graph Types

of size $n \times n$ we choose $\lceil \sqrt{n} \rceil$ centers for both customers and servers by drawing random points for a uniform distribution on the set $[0, n] \times [0, n]$. Each center is assigned a random weight from a $Uniform[0, 1]$ and a 2×2 covariance matrix generated by drawing a random 2×2 matrix with $Uniform[0, 1]$ entrees multiplying it with its transpose and and averaging it with a unit matrix. The workload arrival of customers and service rate of servers then take the values of the $p.d.f$ of the mixture distribution at the node points and values of both are then normalized so that their sum equals to 1. The initial set of edges is the grid connecting the grid nodes so that nodes (x, y) and (x', y') have an edge between them if $||(x - x'), (y - y')|| \leq k$ for some $k \in \mathbb{N}$. However this is likely to lead to system where the CRP condition does not hold and, as will be explained in section 2.1 the system decomposes into subsets of varying workload to service capacity ratios, with some subsets having a ratio greater than 1. As we are interested in feasible systems we divide all workload rates by the maximum workload to service rate ratio amongst all subsets. Therefore, in contrast to the other graph types, the utilization level simulated does not reflect the overall system utilization but rather the utilization of subset of customers classes and servers with the highest workload to service rate ratio.

The Erdős-Rényi, Torus and Map graphs were chosen as each has a different level of natural flexibility in distributing workload across the servers.The homogeneous bipartite Erdős-Rényi is an expander graph (see [30] for formal definition) which for our purposes means that any pair of server nodes in the bipartite graph have multiple short paths between them and hence workload can be naturally transferred between the servers when the system is congested. The k tours graphs are constructed for the CRP condition to hold, however, unlike the Erdős-Rényi graph they are not expander graphs and the avg. length of the shortest path from a node to all other nodes scales linearly with the size of the tours. As a result, even though it is theoretically possible to apply an idling policy that balances the workload across the servers, it is difficult to transfer workload across servers at time of congestion. The map are similar to the tours graph in that the avg. distance between nodes scales with the size of the graph, the difference being that CRP condition is not likely to

hold for Map graphs and hence not only is difficult to transfer workload across the servers at times of congestion, but also doing so may render the system unstable.

Chapter 2

MinMax Fairness and Complete Resource Pooling

2.1 MinMax fairness in SBPSS

Three fundamental questions regarding the structure of a system are addressed in this chapter. The first question to be considered is whether or not the system $\mathcal{F}$ can be stabilized, i.e., does there exist a policy under which the limits in (1.16) exist. A second question of interest is whether or not there exists a policy Ψ that distributes the workload across the capacity of the servers so that the server utilization is uniform across all servers and $\rho_j = \rho$ for some $\rho > 0$ for $j = 1, \ldots, n$. Finally, if there is no policy that can balance the utilization evenly across the servers, how does one define the most *balanced* server utilization distribution that can be achieved and how can it be obtained. The answer to all three questions can be answered by means of a *min-max-fair* set of matching rates. In this section we first introduce the static allocation problem associated with an SBPSS and use it to define the concept of a *min-max-fair* set of matching rates. We then go on to show how a *min-max-fair* set of matching rates $\mathbf{r}^f$ can be used to derive some structural properties of the SBPSS. These properties will be used later in Chapter 4 to construct improved service policies. Finally, we provide a method for obtaining the server utilizations under a *min-max-fair* set matching rates. The notion of a *min-max-fairness* is commonly used in communication networks as an indicator of fairness in the allocation of bandwidth [31]. In order to define *min-max-fairness* in the context of an SBPSS we consider an *input-queued* version of our *output-queued* system. In this paper we focus on an SBPSS that operates under an *input-queued*(IQ) discipline by which the arriving customers are placed in "input" based queues by their respective customer classes. In an *output-queued*(OQ) system arriving customer are routed to a specific server queue immediately upon arrival and are placed in an "output" based queue that belongs to the assigned server. A policy of interest for the OQ system is the non-idling Markovian routing policy in which upon arrival a customer of class i is randomly routed to a server $j \in \partial(i)$ with probability $\frac{r_{ij}}{\lambda_i}$. Each server $j \in \mathcal{J}$ serves the customers in the output queue in

order of arrival without idling. In contrast to the IQ system where the matching rates and server utilizations are a consequence of the matching policy, under Markovian routing a fixed set of matching rates defines the matching policy and the resulting set of server utilizations. Let $\Delta_{\boldsymbol{\lambda},\boldsymbol{\mu}}$ be the set of admissible matching rates given by:

$$\Delta_{\boldsymbol{\lambda},\boldsymbol{\mu}} = \left\{ \mathbf{r} \in \mathbb{R}_+^{m\times n} \left| \sum_{j\in\partial(i)} r_{ij} = \lambda_i, \forall i \in \mathcal{I} \text{ and } \sum_{i\in\partial(j)} r_{ij}\eta_i \leq \mu_j, \forall j \in \mathcal{J} \right. \right\} \tag{2.1}$$

And for a set of admissible matching rates $\mathbf{r}$ the resulting server utilizations are given by

$$\rho_j(\mathbf{r}) = \frac{\sum_{i\in\partial(j)} \lambda_i s_i}{\mu_j} \tag{2.2}$$

Under a Markovian routing policy every individual server and associated queue form a separate $M/M/1$ queue. Therefore, the static planning problem associated with the output queued system with Markovian routing is defined as

$$\min_{\rho\in\mathbb{R}_+, \mathbf{r}\in\mathbb{R}_+^{m\times n}} \rho \tag{2.3}$$

subject to:

$$\sum_{j\in\partial(i)} r_{ij} = \lambda_i, \quad \text{for } i = 1,\ldots,m \tag{2.4}$$

$$\sum_{i\in\partial(j)} r_{ij}\lambda_i s_i \leq \mu_j \rho, \quad \text{for } j = 1,\ldots,n \quad . \tag{2.5}$$

The solution of the above LP itself can answer the first two questions posed. First, we recall that $\boldsymbol{\mu} > 0$ and the graph $G = (\mathcal{I} \cup \mathcal{J}, E)$ is assumed to be connected so that $\partial(i) \neq \emptyset$ and therefore, since the value of ρ is not restricted, the LP in (2.3),(2.4),(2.5) is feasible and a finite strictly positive optimal solution exists. If the optimal value is $\rho^* = \bar{\rho}$ where $\bar{\rho}$ is the average utilization of the system, which is independent of the policy and given by:

$$\bar{\rho} = \frac{\sum_{i\in\mathcal{I}} \lambda_i s_i}{\sum_{j\in\mathcal{J}} \mu_j} \tag{2.6}$$

then all the constraints in (2.5) must hold at equality and a Markovian policy defined by a set of optimal matching rates will induce a uniform utilization across all servers. If in addition $\rho^* < 1$ then the system may be stabilized as employing the Markovian routing policy with the optimal rates $\mathbf{r}$ will cause all queues to operate as independent subcritical $\Sigma M/M/1$ queues. If $\rho^* > \bar{\rho}$ than the workload can not be spread equally across the servers, as doing so will contradict the optimality of ρ^* and we provide the following definition of *min-max-fairness* to characterize the most balanced distribution of server utilizations.

Definition 2.1.1. *A set of admissible matching rates* $\mathbf{r}^f \in \Delta_{\boldsymbol{\lambda},\boldsymbol{\mu}}$ *is said to be min-max-fair if and only if for any other set of admissible matching rates* $\mathbf{r} \in \Delta_{\boldsymbol{\lambda},\boldsymbol{\mu}}$ *the following holds: If there exists a server* $j \in \mathcal{J}$ *such that* $\rho_j(\mathbf{r}) < \rho_j(\mathbf{r}^f)$ *then there must also exist a server* $j' \in \mathcal{J}$ *such that* $\rho_{j'}(\mathbf{r}^f) \geq \rho_j(\mathbf{r}^f)$ *and* $\rho_{j'}(\mathbf{r}) > \rho_{j'}(\mathbf{r}^f)$.

Simply put, an admissible set of matching rates is considered *min-max-fair* if the resulting utilization of any server can not be reduced without increasing the resulting utilization of another server that already has a higher or equal utilization. In the context of this paper we do not make direct use of a *min-max-fair* rates themselves. Instead, we use the unique graph decomposition that, as we will now show, is induced by a *min-max-fair* set of matching rates.

Min Max Fair SBPSS Decomposition

Any admissible set of matching rates $\mathbf{r}$, when used to define a Markovian routing policy, induces a set of $\ell(\mathbf{r})$ unique utilization values. $\bar{\rho}_{\ell(\mathbf{r})}(\mathbf{r}) < \cdots < \bar{\rho}_1(\mathbf{r})$ where $1 \leq \ell(\mathbf{r}) \leq n$. To avoid cumbersome notation we will let $\ell = \ell(\mathbf{r})$ when there is no concern of ambiguity. These distinct utilization values imply a partition of the set of servers $\mathcal{J}$ into subsets $J_1(\mathbf{r}), \ldots, J_\ell(\mathbf{r})$ so that all servers in the same subset have the same utilization under matching rates $\mathbf{r}$. That is, for all $j \in J_k(\mathbf{r}), \rho_j(\mathbf{r}) = \bar{\rho}_k(\mathbf{r})$. For an admissible set of matching rates $\mathbf{r}$, a server $j \in \mathcal{J}$ is said to *actively serve* customer class $i \in \partial(j)$ if and only if $r_{ij} > 0$. Hence, the partition of the servers, $J_1(\mathbf{r}), \ldots, J_\ell(\mathbf{r})$, implies a collection of ℓ subsets of actively served customer classes, $I_1(\mathbf{r}), \ldots, I_\ell(\mathbf{r})$, where

$$I_k(\mathbf{r}) = \{i \in I | r_{ij} > 0, j \in J_k(\mathbf{r})\}, \quad k = 1, \ldots, \ell \tag{2.7}$$

is the set of customers that are actively served by a server in $J_k(\mathbf{r})$. Clearly, for an admissible $\mathbf{r}$, the union of the sets $I_1(\mathbf{r}), \ldots, I_\ell(\mathbf{r})$ includes all the customers so that, $\cup_{k=1}^{\ell} I_k(\pi) = \mathcal{I}$, as otherwise $\mathbf{r}$ could not be admissible. We now state the main theorem of this Chapter.

Theorem 2.1.1. *A set of matching rates* $\mathbf{r}$ *is min-max-fair if and only if it is feasible and* $(I_p(\mathbf{r}) \times J_q(\mathbf{r})) \cap E = \emptyset$ *for any pair* p, q *where* $1 \leq p < q \leq \ell(\pi)$

Before we turn to proving the theorem we first present the following lemma that establishes the connection between the compatibility graph structure and the *min-max-fair* matching rates.

Lemma 2.1.1. *Let* $\mathbf{r}$ *be an admissible set of matching rates with a utilization-based partition of servers,* $J_1(\mathbf{r}), \ldots, J_\ell(\mathbf{r})$*, and induced subsets of actively served customers,* $I_1(\mathbf{r}), \ldots, I_\ell(\mathbf{r})$ *such that* $(I_p(\mathbf{r}) \times J_q(\mathbf{r})) \cap E = \emptyset$ *for any pair* p, q *where* $1 \leq p < q \leq \ell$*. Then:*

1. *The sets* $I_1(\mathbf{r}), \ldots, I_\ell(\mathbf{r})$ *are mutually disjoint and form a partition of* $\mathcal{I}$.
2. $\partial(I_p(\mathbf{r})) \subseteq \cup_{k=1}^{p} J_k(\mathbf{r}), \quad \mathbf{r}$ *for* $p = 1, \ldots, \ell$
3. $\partial(J_p(\pi)) \subseteq \cup_{k=p}^{\ell} I_k(\mathbf{r}), \quad \mathbf{r}$ *for* $i = k, \ldots, \ell$

Proof. (1). By contradiction, assume that $\mathbf{r}$ is admissible and that $I_1(\mathbf{r}), \ldots, I_\ell(\mathbf{r})$ do not form a partition of $\mathcal{I}$. For any admissible $\mathbf{r} \in \Delta_{\boldsymbol{\lambda},\boldsymbol{\mu}}$ we know $\cup_{k=1}^{\ell} I_k(\mathbf{r}) = \mathcal{I}$ and hence in order for $I_1(\mathbf{r}), \ldots, I_\ell(\mathbf{r})$ to not form a partition there must exist some $1 \leq p < q \leq \ell$ such that $I_p(\mathbf{r}) \cap I_q(\mathbf{r}) \neq \emptyset$. Let $i \in I_p(\mathbf{r}) \cap I_q(\mathbf{r})$ then by definition i is actively served by at least one server in both $J_p(\mathbf{r})$ and $J_q(\mathbf{r})$ and hence there must exist a qualification between a customer $i \in I_p(\mathbf{r})$ and server $j \in J_q(\mathbf{r})$ in contradiction to our initial assumption. **(2)** Let $\mathbf{r}$ be a feasible . If for some $1 \leq p \leq \ell$, $J(I_p(\mathbf{r})) \not\subseteq \cup_{k=1}^{p} J_k(\mathbf{r})$ then there must exist some $q > p$ such that $J(I_p(\mathbf{r})) \cap J_q \neq \emptyset$ and hence there must be an arc from J_q to I_p which is a contradiction. **(3)** Similarly, if for some $1 \leq p < \ell$ we have $I(J_p(\mathbf{r})) \not\subseteq \cup_{k=p}^{\ell} I_k(\mathbf{r})$ then there must exist some $q < p$ such that $I(J_p(\mathbf{r})) \cap I_q(\mathbf{r}) \neq \emptyset$ and hence there is an arc from $I_q(\mathbf{r})$ to $J_p(\mathbf{r})$ which is a contradiction . $\square$

We are now ready to prove theorem 2.1.1

. **Proof of Theorem 2.1.1:** Let $\mathbf{r}$ be an admissible set of matching rates with a utilization-based partition of servers $J_1(\mathbf{r}), \ldots, J_\ell(\mathbf{r})$, corresponding utilization sequence $\bar{\rho}_1(\mathbf{r}) > \cdots > \bar{\rho}_\ell(\mathbf{r})$ and induced subsets of actively served customers, $I_1(\mathbf{r}), \ldots, I_\ell(\mathbf{r})$. First, let us assume that there exists a qualification $(i,j) \in I_p(\mathbf{r}) \times J_q(\mathbf{r}) \cap E$ for some $1 \leq p < q \leq \ell$. Since $i \in I_p(\pi)$ there must exist some server $j' \in J_p(\mathbf{r})$ such that $r_{ij'} > 0$ and $\bar{\rho}_p(\mathbf{r}) = \rho_{j'}(\mathbf{r}) > \rho_j(\mathbf{r}) = \bar{\rho}_q(\mathbf{r})$. Hence, let

$$\epsilon = min\{(\bar{\rho}_p(\mathbf{r}) - \bar{\rho}_q(\mathbf{r}))\frac{\mu_j \mu_{j'}}{2(\mu_j + \mu_{j'})}, \pi_{ij'}\} > 0 \tag{2.8}$$

and let $\mathbf{r}'$ be an such that

$$\mathbf{r}'_{uv} = \begin{cases} \mathbf{r}_{ij'} - \epsilon & \text{if } u = i, v = j' \\ \mathbf{r}_{ij} + \epsilon & \text{if } u = i, v = j \\ \mathbf{r}_{uv} & \text{otherwise} \end{cases} .$$

The fact that $\epsilon > 0$ implies that both $\rho_j(\mathbf{r}') > \rho_j(\mathbf{r})$ and $\rho_{j'}(\mathbf{r}') < \rho_{j'}(\mathbf{r})$ and since $\mathbf{r}$ was assumed feasible we can observe that:

$$\rho_{j'}(\mathbf{r}') - \rho_j(\mathbf{r}') = \rho_{j'}(\mathbf{r}) - \frac{\epsilon}{\mu_{j'}} - (\rho_j(\mathbf{r}) + \frac{\epsilon}{\mu_j}) =$$
$$\rho_{j'}(\mathbf{r}) - \rho_j(\mathbf{r}) - \epsilon\frac{\mu_j + \mu_{j'}}{\mu_j \mu_{j'}} \geq \frac{\rho_{j'}(\mathbf{r}) - \rho_j(\mathbf{r})}{2} > 0$$

and hence $0 < \rho_j(\mathbf{r}') < \rho_{j'}(\mathbf{r}') < 1$ and $\mathbf{r}'$ is feasible as well. Therefore, we have $\rho_{j'}(\mathbf{r}') < \rho_{j'}(\mathbf{r})$ and for any server $h \in \cup_{k=1}^{p} J_k(\mathbf{r})$ we have $\rho_h(\mathbf{r}') \leq \rho_h(\mathbf{r})$ and hence the matching rates $\mathbf{r}$ are not *min-max-fair*.

To prove the converse let us now assume that $I_p(\mathbf{r}) \times J_q(\mathbf{r}) \cap E = \emptyset$ for any pair p, q where $1 \leq p < q \leq \ell$. Let $\mathbf{r}' \in \Delta_{\boldsymbol{\lambda},\boldsymbol{\mu}}$ be a set of matching rates such that there exists some $j \in \mathcal{J}$ for which $\rho_j(\mathbf{r}') < \rho_j(\mathbf{r})$. We will now show that there must also exist some j' for which $\rho_j(\mathbf{r}) \leq \rho_{j'}(\mathbf{r}) < \rho_{j'}(\mathbf{r}')$. Let p be such that $j \in J_p(\mathbf{r})$ and so $\rho_j(\mathbf{r}) = \bar{\rho}_p(\mathbf{r})$. From Lemma

2.1.1 we know that the sets $I_1(\mathbf{r}), \ldots, I_\ell(\mathbf{r})$ form a partition of $\mathcal{I}$ and therefore

$$\sum_{k=1}^{p} \sum_{i \in I_k(\mathbf{r})} \lambda_i s_i = \sum_{k=1}^{p} \sum_{v \in S_k(\mathbf{r})} \mu_v \rho_k(\mathbf{r}). \tag{2.9}$$

Lemma 2.1.1 also states that $J(I_p(\pi)) \subseteq \cup_{k=1}^{p} J_k(\pi)$ and hence, since $\mathbf{r}'$ is feasible, we have:

$$\sum_{k=1}^{p} \sum_{i \in I_k(\mathbf{r})} \lambda_i s_i \leq \sum_{k=1}^{p} \sum_{v \in J_k(\mathbf{r})} \mu_v \rho_k(\mathbf{r}') \tag{2.10}$$

Therefore, since $\rho_j(\mathbf{r}') < \rho_j(\mathbf{r})$ we must have

$$\sum_{k=1}^{p} \sum_{\substack{v \in J_k(\mathbf{r}) \\ v \neq j}} \mu_v \rho_k(\mathbf{r}) < \sum_{k=1}^{p} \sum_{\substack{v \in J_k(\mathbf{r}) \\ v \neq j}} \mu_v \rho_k(\mathbf{r}')$$

and hence there must exist some $j' \in \cup_{k=1}^{p} J_k(\mathbf{r}) \setminus \{j\}$ such that $\rho_{j'}(\mathbf{r}') > \rho_{j'}(\mathbf{r})$ and, since $j' \in \cup_{k=1}^{p} J_k(\mathbf{r})$, $\rho_{j'}(\mathbf{r}) \geq \rho_j(\mathbf{r})$ and therefore $\mathbf{r}$ is *min-max-fair* $\square$.
Having proven this property of the server utilizations induced by a set of *min-max-fair* matching rates we now wish to characterize the set of all *min-max-fair* matching rates by their induced server utilizations.

Theorem 2.1.2. *For any given SBPSS there exist unique partitions $J_1, \ldots, J_\ell$ and $I_1, \ldots, I_\ell$ of the server set $\mathcal{J}$ and customer set $\mathcal{I}$ respectively such that a set of admissible matching rates $\mathbf{r} \in \Delta_{\boldsymbol{\lambda},\boldsymbol{\mu}}$ is min-max-fair if and only if $J_k(\mathbf{r}) = J_k, I_k(\mathbf{r}) = I_k$ for $k = 1, \ldots, \ell$*

Proof Let $\mathbf{r}, \mathbf{r}' \in \Delta_{\boldsymbol{\lambda},\boldsymbol{\mu}}$ both be *min-max-fair*. By definition of *min-max-fairness* both $\mathbf{r}$ and $\mathbf{r}'$ must obtain the minimal maximum utilization of the system, otherwise any other set of matching rates that does achieve the maximum minimum utilization, which must exist by definition, would constitute a contradiction to the *min-max-fairness* of $\mathbf{r}, \mathbf{r}'$. Therefore we can conclude that $\bar{\rho}_1(\mathbf{r}) = \bar{\rho}_1(\mathbf{r}')$. If we now assume that $J_1(\mathbf{r}) \neq J_1(\mathbf{r}')$ and that,*w.l.o.g*, $J_1(\mathbf{r}) \setminus J_1(\mathbf{r}') \neq \emptyset$ then there exists a server $j \in J_1(\mathbf{r})$ such that $\rho_j(\mathbf{r}) \neq \rho_j(\mathbf{r}')$. The utilization of any server in $J_1(\mathbf{r}) \setminus J_1(\mathbf{r}')$ under $\mathbf{r}'$ must, by definition, be strictly smaller than $\bar{\rho}_1(\mathbf{r}') = \bar{\rho}_1(\mathbf{r})$ and hence:

$$\sum_{v \in J_1(\mathbf{r})} \rho_v(\mathbf{r}') \mu_v < \sum_{v \in J_1(\mathbf{r})} \rho_j(\mathbf{r}) \mu_v$$

The sets $I_1(\mathbf{r}), \ldots, I_{\ell(\mathbf{r})}(\mathbf{r})$ are a partition of $\mathcal{I}$ by Lemma 2.1.1 and therefore

$$\sum_{v \in S_1(\mathbf{r})} \rho_v(\mathbf{r}) \mu_v = \sum_{i \in I_1(\mathbf{r})} \lambda_i$$

However, lemma 2.1.1 also states that $J(I_1(\mathbf{r})) = J_1(\mathbf{r})$ and therefore

$$\sum_{v \in J(I_1(\mathbf{r}))} \mu_v \rho_v(\mathbf{r}') < \sum_{i \in I_1(\mathbf{r})} \lambda_i$$

which is a contradiction to the feasibility of $\mathbf{r}'$ and we conclude that $J_1(\mathbf{r}) = J_1(\mathbf{r}')$. The equality of the sets $I_1(\mathbf{r}) = I_1(\mathbf{r}')$ follows immediately since by Theorem 2.1.1 these are the sets of customers who may only be served by servers in $J_1(\mathbf{r}) = J_1(\mathbf{r}')$. We have therefore, shown that any pair of *min-max-fair* matching rates $\mathbf{r}, \mathbf{r}'$ share the same maximum utilization value and the respective set of servers and customers classes, $J_1(\mathbf{r})$, $I_1(\mathbf{r})$. Lemma 2.1.1 implies that for a set of *min-max-fair* matching rates the servers in $J_1(\mathbf{r})$ can only actively serve the customers in $I_1(\mathbf{r})$ and hence we can remove the sets $J_1(\mathbf{r}), I_1(\mathbf{r})$ with their adjacent qualifications from the SBPSS. The matching rates on the remaining SBPSS, which remain unchanged, still meet the conditions of Theorem 2.1.1 and therefore form a set *min-max fair* matching rates with respect to the sets $\mathcal{I} \setminus I_1(\mathbf{r}), \mathcal{J} \setminus J_1(\mathbf{r})$, if those are not empty, and the same argument can be repeated to prove that $J_2(\mathbf{r}) = J_2(\mathbf{r}')$ and $I_2(\mathbf{r}) = I_2(\mathbf{r}')$. Repeating the same argument ℓ times we can conclude that all *min-max-fair* matching induce the same sequence $\rho_1, \ldots, \rho_\ell$ of utilization values and the same set partitions J_k, I_k for $k = 1, \ldots, \ell$. $\square$.

An immediate corollary of the theorem is:

Corollary 2.1.1. *There exists a unique utilization sequence $\bar{\rho}_1, \ldots, \bar{\rho}_\ell$ such that*

$$\bar{\rho}_k = \frac{\sum_{i \in I_k} \lambda_i}{\sum_{j \in J_k} \mu_j}, \quad k = 1, \ldots, \ell \tag{2.11}$$

and for any set of min-max-fair matching rates $\mathbf{r}$ and $1 \leq k \leq \ell$,

$$\rho_j(\mathbf{r}) = \bar{\rho}_k \Leftrightarrow j \in J_k \tag{2.12}$$

.

Theorem 2.1.2 and subsequent corollary 2.1.1 provide us with a decomposition of the SBPSS into ℓ subsystems that, for any set of *min-max-fair* matching rates, have equal utilizations across the subsystem servers. Furthermore, the *min-max-fairness* implies that the utilization of any server in a subsystem may not be reduced without increasing the utilization of a server of similar or possibly higher utilization. In the remainder of the paper, we will refer to these sets and utilization values as the *min-max-fair* decomposition of the system $\mathcal{F}$. Finally, let $\phi : \mathcal{I} \cup \mathcal{J} \to \{1, \ldots, \ell\}$ be a mapping of the system's customer classes and servers to their respective *min-max-fair* utilization sets such that:

$$\forall v \in \mathcal{I} \cup \mathcal{J}, \phi(v) = k \Leftrightarrow v \in I_k \cup J_k. \tag{2.13}$$

Deriving the Min Max Fair Decomposition

Now that the existence of the unique fair decomposition has been established we present an algorithm for deriving the unique decomposition using a parametric minimum cut procedure on a flow graph model of the fluid SBPSS. Let us consider a fluid model of a SBPSS represented by a bipartite parametric flow graph. Let the customer set and server sets $\mathcal{I}, \mathcal{J}$ be

represented by the bipartite node sets in the graph and let the set of compatible customer-server pairs define the edge set E with an infinite flow upper bound. Let us now add two auxiliary nodes s, t representing a source and terminal for the graph flow. For each customer class $i \in \mathcal{I}$ we add an edge with a capacity upper bound of $\eta_i = \lambda_i s_i$, the rate at which customers of class i introduce workload to the systems, connecting node i to the source node s and we denote by $E_s = \{(s, i) | i \in \mathcal{I}\}$ the set of source adjacent edges. Similarly, for each server $j \in \mathcal{J}$ we add an edge with capacity upper bound of $\mu_j \cdot \rho$ connecting node j to the destination node t and we denote by $E_t = \{(j, t) | j \in \mathcal{J}\}$ the set of terminal adjacent edges. This results in a parametric bipartite flow graph $G(\rho) = (\{s\} \cup \mathcal{I} \cup \mathcal{J} \cup \{t\}, E_s \cup E \cup E_t)$ such

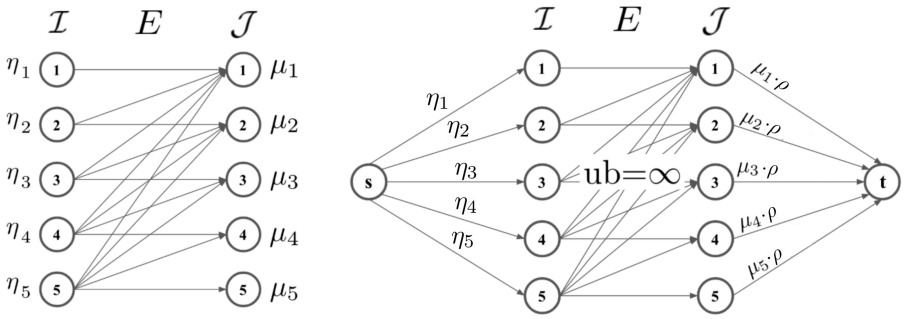


Figure 2.1: Transformation of the bipartite compatibility graph to a parametric $s - t$ flow graph with parameter ρ

as the one in Figure 2.1 where the utilization parameter $\rho \geq 0$ determines the maximum server utilization. An $s - t$ flow in graph $G(\rho)$ is a vector $\boldsymbol{f} \in \mathbb{R}^{m+n+|E|}$. A flow in graph $G(\rho)$ is a feasible flow if the flow across any edge does not exceed the upper bound of the edge. A flow vector $\boldsymbol{f}$ on $G(\rho)$ is said to be *source saturating* if and only if,

$$f_{si} = \sum_{j \in \partial(i)} f_{ij} = \eta_i \tag{2.14}$$

meaning that a flow vector can be translated into an admissible set of matching rates if and only if it saturates all arcs adjacent to the origin node. Clearly, the set of admissible flow vectors $G(\rho)$ is bijective to the set of admissible matching rates which induce a maximum utilization rate of smaller or equal to ρ with the bijection given by

$$r_{ij} = \frac{f_{ij}}{s_i}, \quad (i, j) \in E. \tag{2.15}$$

The utilization of a server $j \in \mathcal{J}$ induced by an flow vector $\boldsymbol{f}$ is given by

$$\rho_j(\boldsymbol{f}) = \frac{1}{\mu_j} \sum_{i \in \partial(j)} f_{ij}. \tag{2.16}$$

A parametric minimum cut algorithm such as the ones descried in Section 4.1 of [23] and in [29](the latter being the one used for the experiments in this paper see also [16], [21]) takes as input the parametric graph $G(\rho)$ and a range $\rho \in [0, \rho_0]$ and returns a set of breakpoints $\rho_0 \geq \rho_1 \geq \cdots \geq \rho_\ell \geq 0$ and respective $s-t$ cuts $(X_1, \bar{X}_1), \ldots, (X_\ell, \bar{X}_\ell)$ where X_k is the minimal source set of the minimum cut in the graph $G(\rho)$ for $\rho_{k-1} < \rho \leq \rho_k$. Let $\mu_{\min}^{-1}\rho_0 = \sum_{i\in\mathcal{I}} \lambda_i s_i$ where $\mu_{\min} = min\{\mu_j, j \in \mathcal{J}\}$ so that all the workload of the system can be handled by the slowest server when $\rho = \rho_0$ and we are guaranteed that all the source adjacent arcs are saturated by a maximum flow and the minimal source set of the minimum cut in $G(\rho_0)$ is $\{s\}$. To avoid cumbersome notation we denote $X_k \cap \mathcal{J}$ by $X_k^{(\mathcal{J})}$. Let us now consider the minimum source set at the a break point ρ_k. First we observe that for any $(i,j) \in (\bar{X}_k^{(\mathcal{I})} \times X_k^{(\mathcal{J})}) \cap E$ we must have $f_{ij} = 0$, otherwise for any $i' \in \partial(j) \cap X_k^{(\mathcal{I})}$, which must exist as the arc (j,t) is saturated, the path $i' - j' - i$ would be an augmenting path. Furthermore, we must have $(X_k^{(\mathcal{I})} \times \bar{X}_k^{(\mathcal{J})}) \cap E = \emptyset$ otherwise $(X_k, \bar{X}_k)$ would not be a minimum cut as $ub_{ij} = \infty$ for any $(i,j) \in E$. Therefore, at the $k-th$ breakpoint we have a flow assignment where the flow out of customer nodes of $X_k \cap \mathcal{I}$ is only assigned to server nodes of $X_k \cap \mathcal{J}$, the servers of $x_k \cap \mathcal{J}$ are assigned no flow from the nodes in $\bar{X}_k \cap \mathcal{I}$ and the arcs of $E_t \cap (X_k \cap \mathcal{J}) \times \{t\}$ are fully saturated. Theorem 2.1.2 implies that

$$X_k^{(\mathcal{I})} = \bigcup_{q=1}^{k} I_q \quad \text{and} \quad X_k^{(\mathcal{J})} = \bigcup_{q=1}^{k} J_q \tag{2.17}$$

where $I_q, J_q, q = 1, \ldots, k$ are the respective customer and server sets of the unique *min-max-fair* decomposition. Therefore, as this applies to any $k = 1, \ldots \ell$, we can retrieve the decomposition from the set of source cuts as by applying

$$I_q = X_q^{(\mathcal{I})} \setminus X_{q-1}^{(\mathcal{I})} \quad \text{and} \quad X_q^{(\mathcal{J})} \setminus X_{q-1}^{(\mathcal{J})} \quad q = 1, \ldots \tag{2.18}$$

and the *min-max-fair* utilization sequence is the set of breakpoint values. A full *min-max-fair* assignment can then be obtained by setting the following upper bounds

$$ub_{jt}^{f} = argmax\{\rho_k | j \in X_k\} \tag{2.19}$$

on the edges of E_t and solving a max-flow problem. The resulting flow vector $\boldsymbol{f}^*$ must induce the set of *min-max-fair* utilization rates and hence the corresponding set of matching rates is a *min-max-fair*

2.2 Complete Resource Pooling

The complete resource pooling condition, first introduced by [28] and [55] defines the conditions under which a general SBPSS, not necessarily with homogeneous service, can be operated as a single pooled resource. For a general SBPSS the amount of workload required to serve a customer class $i \in \mathcal{I}$ depends on the service policy and subsequent matching

rates. Therefore, the resource pooling condition is defined only for arrival rates that must utilize the full capacity of the servers in order to process the arriving workload. In an SBPSS with homogeneous service the rate of workload arrival to the system is independent of the policy, hence we provide definitions of the complete resource condition for a system with homogeneous service

Definition 2.2.1 (Complete Resource Pooling)**.** *An SBPSS* $\mathcal{F} = (\mathcal{I} \cup \mathcal{J}, E, \boldsymbol{\lambda}, \mathbf{s}, \boldsymbol{\mu})$ *is said to satisfy the complete resource pooling(CRP) condition if and only if*

$$\frac{\sum_{i \in I} \lambda_i s_i}{\sum_j \in \partial(I) \mu_j} \leq \frac{\sum_{i \in mI} \lambda_i s_i}{\sum_{j \in \mathcal{J}} \mu}, \quad \forall I \subsetneq \mathcal{I} \tag{2.20}$$

If (2.20) holds a strict inequality for all $I \subsetneq \mathcal{I}$ then system is said to satisfy the *strong*-CRP condition otherwise, it satisfies the *weak*-CRP condition. Both the weak and strong CRP conditions guarantee that utilization can be balanced across the servers and that the *min-max-fair* decomposition is trivial with $I_1 = \mathcal{I}$. However, if the system satisfies *weak*-CRP but not the *strong*-CRP then for any set of *min-max-fair* matching rates $\mathbf{r}^f$ that balance the utilization the graph restricted to active arcs $E_{\mathbf{r}^f} = \{(i,j) | r^f_{ij} > 0, (i,j) \in E\}$ is disconnected and hence the system can only be balanced if it is decomposed into subsystems of smaller size and equal utilization.

Chapter 3

Matching Rate Approximations

3.1 Max. Entropy Approximation of Infinite FCFS Sequence Matching Rates

Literature Review - Match Rate Approximation

In the FCFS stochastic matching model one is given two random infinite sequences of customers and servers. As in a SBPSS model, the customers and servers are of classes $\mathcal{I} = \{1, \ldots, m\}$ and $\mathcal{J} = \{1', \ldots, n'\}$ respectively and a bipartite graph $G = (\mathcal{I} \cup \mathcal{J}, E)$ defines the compatible customer-server pairs. The class of every customer or server in the respective sequences is an $i.i.d$ random variable that assumes the value $i \in \mathcal{I}$ with probability α_i in the case of the customer sequence and a value $j \in \mathcal{J}$ with probability β_j in case of the sever sequence with $||\boldsymbol{\alpha}||_1 = ||\boldsymbol{\beta}||_1 = 1$. The FCFS matching procedure works as follows: The customer of class $i \in \mathcal{I}$ at the first position in the customer stream scans the server stream until it finds the first compatible server $j \in \partial(i)$, the match is logged as an (i, j) match and both customer and server are removed from the stream. After the match is executed, all upstream customers and servers move one position forward. The matching process can be performed in various ways, for example one can iterate over the servers instead of over the customers. Let r_{ij}^k be the count of (i, j) matches after a total of k matches have been made. The matching rates of the infinite FCFS matching sequence are defined as:

$$r_{ij} = \lim_{k \to \infty} \frac{r_{ij}^k}{k}. \tag{3.1}$$

The FCFS infinite matching sequence can be used to model various applications in which both the supply and demand for a resource arrive randomly to the system and the resource is allocated to the earliest compatible demand. The model was first introduced in [34] for the purpose of analyzing the allocation of public housing to applicants in the city of Boston, MA. In [3] the authors suggest using an FCFS infinite matching sequence to model organ transplant and adoption procedures and in more recent work [1] the authors show the relation

between the FCFS infinite matching sequence model and various other queueing models such as redundancy models [24] wherein multiple copies of a customer are sent to different queues to reduce latency and arriving servers leave the system when no compatible customers are in the queue. The connection between the infinite matching model and an SBPSS was first suggested in [13] where the authors observed that in heavy traffic the matching rates of an SBPSS operating under a FCFS server policy converges to those of an infinite matching sequence with corresponding customer and server frequencies. The authors than go on to propose the first approximation scheme for the infinite matching sequence by means of an iterative procedure that at each iteration allocates the unmatched portion of arrival/service of each customer/server node proportionally across its set of neighbors. The matching rates for an overloaded system where the arrival rates of customers exceed the capacity of the servers and the customers become impatient was derived by [48] for certain graph topologies. The quasi-independent approximation of the matching rates is first described in [14] based on a physical analogy of the system to a network of pipelines conducting a flow of non-Newtonian fluids with pipes as edges and pressure exerted at nodes. The approximation we will introduce in Section 3.1 is identical to the approximation of [14]. Closed-form expressions for the exact matching rates of an infinite FCFS matching sequence were derived using a novel Markov chain formulation (originally due to [53]) in a seminal paper [3]. Unfortunately, the closed-form expressions in [3] require calculation of a distinct term for every permutation of the server set and hence can only used to obtain the matching rates of relatively small systems (12 nodes on each side according to [3]). Extending upon the steady state probabilities derived in [3] the authors derive closed form expressions for the stationary distribution of an SBPSS operating under the FCFS-ALIS policy, (see the subsequent subsection for a full description). Recently, in [20] inspired by similarities between the closed-form matching rates of [3] and Ohm's Law the authors suggested an approximation of the infinite FCFS bipartite matching [20] that is based on an analogous electrical circuit. They provide some empirical results for the approximation, (see the following subsections for a detailed description of the approximation). However, in an even more recent paper, [4] the authors demonstrate that the Ohm's Law based approximation of [20] may produce negative approximations of the matching rates. They propose an improved approximation based on an electrical circuit that contains diodes to prevent the negative current flows. The approximation requires the solution of a quadratic rather than the linear program of [20] and so we shall refer to it as the the QP approximation. The authors in [4] prove that their suggested QP approximation coincides with the Ohm's Law approximation whenever the latter does produce a valid (i.e positive) approximation. In the remainder of this Chapter we will first introduce the closed form stationary distribution and matching rates of [3], [2] and then go on to re-derive the quasi-independent approximation of [14] from basic principles as a Maximum Entropy approximation. Having introduced the approximation we then describe a surprising and enlightening counter example. The accuracy of approximations is then tested by repeating the experiments of [20] which show the Maximum Entropy approximation is more accurate and robust than the QP and Ohm's Law approximations, especially for sparse graphs. We conclude the section by demonstrating the efficacy of the approximation on the large scale

sparse models of section 1.5

The Adan-Weiss Exact Match Rate Equations

The analysis presented in subsequent subsections relies on the system state representation and stationary distribution of a SBPSS with server dependent service times operating under the FCFS-ALIS policy presented in [2] and the closely related FCFS infinite matching sequence in [3]. Therefore, at this point we provide an introduction to the concepts in the aforementioned works to be referenced in upcoming subsections. Although the work on infinite FCFS matching sequences in [3] predates the work on SBPSS under a FCFS-ALIS policy in [2], the matching process of an SBPSS operating under a FCFS-ALIS, in heavy traffic, given certain conditions are satisfied, converges to those an infinite FCFS matching sequence. Hence, we will first introduce the SBPSS model under a FCFS-ALIS policy and only later introduce the infinite FCFS matching sequence as a limiting case. The first concept we introduce is the system state representation. The state of a system at time t can be represented by a tuple comprised of three types of elements $(\mathfrak{s}, k, \mathbf{v})$ where:

1. $\mathfrak{s} = (s_1, \ldots, s_k, s_{k+1}, \ldots, s_n) \in \mathfrak{P}_{\mathcal{J}}$, where is an n-vector representing a permutation of the server set $\mathcal{J}$ and $\mathfrak{P}_{\mathcal{J}}$ denotes the set off all permutations of the set $\mathcal{J}$. The order of the busy servers is determined by the order of arrival of the customers being served, so that for $1 \leq \ell_1 < \ell_2 \leq k$ the customer being served by server s_{ℓ_1} must have arrived before the customer being served by s_{ℓ_2}. The idle servers are ordered in decreasing order of the time in which they went idle so that server s_n was the first of servers $s_{k+1}, \ldots, s_n$ to become idle and s_{k+1} was the last.

2. $k \in \{0, \ldots, n\}$ is an integer specifying the index of the last busy server in the permutation $\mathfrak{s}$. In case all servers in the system are idle then $k = 0$ and when they are all busy $k = n$

3. $\mathbf{v} \in \mathbb{N}_0^{n \times 1}$ is an integer vector counting the number of waiting customers that arrived between every pair of customers that are currently being served. Hence n_k represents the number of waiting customers that arrived after the customer being served by server s_k, the last busy server in the permutation, and for any $\ell = 1, \ldots, k-1$, n_ℓ is the number of customers that arrived between the arrival of the customer currently being served by server s_ℓ and the customer currently being served by server $s_{\ell+1}$. We follow the convention that if k is specified as the last busy server than all entries of $\mathbf{v}$ starting from the $k+1$-th position must be 0. We refer to the customers waiting in between servers s_ℓ and $s_{\ell+1}$ as customers in the ℓ-th slot.

Note that given an arbitrary state $(\mathfrak{s}, k, \mathbf{v})$ there are $k + \sum_{\ell=1}^{k} n_\ell$ customers in the system, out of which k are being served by k busy servers while $n - k$ servers are idle. An important property of the state space representation of [2] is that it does not maintain a count the number of customers by class. Instead, the FCFS-ALIS property implies that the customers

in the ℓ-th slot were skipped by all servers $s_{\ell+1}, \ldots, s_k$ and hence can only be served by servers $s_1, \ldots, s_\ell$. Otherwise it would imply that a server had "skipped" a compatible customer and violated the FCFS-ALIS policy. This novel *opaque* representation of the system state was first suggested by [53] and is the key element that enables the derivation of the steady state probabilities and exact matching rates. Given a permutation $\mathfrak{s} \in \mathfrak{P}_{\mathcal{J}}$ and an integer $\ell \in \{1, \ldots, n\}$ we let $\mathfrak{s}_\ell = \{s_1, \ldots, s_\ell\}$ denote the first ℓ servers in the permutation $\mathfrak{s}$. For a set $J \subseteq \mathcal{J}$ we denote by $\mathcal{U}(J)$ the set of all customer classes that are only served by servers in the set J:

$$\mathcal{U}(J) = \mathcal{I} \setminus \partial(\mathcal{J} \setminus J). \tag{3.2}$$

Given a permutation $\mathfrak{s} \in \mathfrak{P}_{\mathcal{J}}$ and an integer $\ell = 1, \ldots, n$ we define:

$$\lambda_{(\ell,\mathfrak{s})} = \sum_{i \in \mathcal{U}(\mathfrak{s}_\ell)} \lambda_i, \quad \bar{\lambda}_{(\ell,\mathfrak{s})} = \sum_{i \in \mathcal{I} \setminus \mathcal{U}(\mathfrak{s}_\ell)} \lambda_i, \quad \mu_{(\ell,\mathfrak{s})} = \sum_{j \in \mathfrak{s}_\ell} \mu_j. \tag{3.3}$$

As in [3], we abbreviate notation by removing the indicator of the permutation when there is no concern of ambiguity. The stationary distribution of the system derived in [2] is given by:

$$\pi(\mathfrak{s}, k, \mathbf{v}) = B \cdot \prod_{\ell=1}^{k} \frac{\lambda_{(\ell)}^{n_\ell}}{\mu_{(\ell)}^{n_\ell+1}} \prod_{\ell=k+1}^{n} \bar{\lambda}_{(\ell)}^{-1} \tag{3.4}$$

where B is the normalizing constant obtained by summing over all steady state probabilities. As was shown by [2], if the CRP condition of (2.20) holds, then under a heavy traffic regime with $\rho \to 1$ the matching rates of an SBPSS become equivalent to those of an infinite matching sequence with customer and server frequencies $\boldsymbol{\alpha} = \boldsymbol{\lambda}$ and $\boldsymbol{\beta} = \boldsymbol{\mu}$. The steady state probability of any state with idle servers ($k < n$) vanishes and the steady state probabilities can be rewritten as:

$$\pi(\mathfrak{s}, \mathbf{v}) = B \cdot \prod_{\ell=1}^{n} \frac{\alpha_{(\ell)}^{n_\ell}}{\beta_{(\ell)}^{n_\ell+1}}. \tag{3.5}$$

The matching rates of an infinite matching sequence are derived in [3] by conditioning on the server permutation and summing all matching probabilities given the permutation, which leads to the following closed-form expression for the matching rates:

$$r_{ij} = \alpha_i \beta_j \sum_{\mathfrak{s} \in \mathfrak{P}_{\mathcal{J}}} \sum_{k=1}^{n} \phi_{i,(k)} \prod_{\ell=1}^{k} \left(\beta_{(\ell)} - \chi_{j,(\ell)}\right)^{-1} \prod_{\ell=k}^{J-1} \left(\beta_{(k)} - \alpha_{(k)}\right)^{-1}, \tag{3.6}$$

where

$$\phi_{i,\mathfrak{s},(k)} = \begin{cases} 1 & \textit{if } i \in \mathcal{U}(\mathfrak{s}_k) \\ 0 & \textit{Otherwise} \end{cases} \tag{3.7}$$

and

$$\chi_{j,\mathfrak{s},(k)} = \sum_{u \in \mathcal{U}(\mathfrak{s}_k) \setminus \partial(\mathrm{j})} \alpha_u. \tag{3.8}$$

Here, $\phi_{i,\mathfrak{s},(k)}$ is an indicator of whether or not server i is uniquely served by the servers of $\mathfrak{s}_k$ and may be matched by a server at the k-th slot, and $\chi_{j,\mathfrak{s},(\ell)}$ is the sum of the arrival rates for all customer classes that are uniquely served by the servers in $\mathfrak{s}_k$ but are not served by server j. We abbreviate both by writing $\phi_{i,(k)}$ and $\chi_{i,(l)}$ where the permutation $\mathfrak{s}$ is clear from the context.

The Maximum Entropy Approximation

The Maximum Entropy Approximation of the matching rates is, as we will show, identical in practice to the Quasi-Independent Approximation suggested in [14]. It is given by the solution of the following convex optimization problem:

$$\max \mathcal{H}(\mathbf{r}) = -\sum_{(i,j)\in E} r_{ij} log(r_{ij}) \tag{3.9}$$

subject to:

$$\sum_{j\in\partial(i)} r_{ij} = \alpha_i, \quad for\ i = 1,\ldots,m \tag{3.10}$$

$$\sum_{i\in\partial(j)} r_{ij} = \beta_j, \quad for\ j = 1,\ldots,n \tag{3.11}$$

where $r_{ij}, (i,j) \in E$ is the approximated matching rate between customers of class i and servers of class j. If we assume *w.l.o.g* that both $\alpha_1,\ldots,\alpha_m$ and $\beta_1,\ldots,\beta_n$ sum to 1, the resulting matching rates are the maximum entropy distribution of the flow across the arcs of the compatibility graph. The maximum entropy function is 1-strongly concave in the ℓ_1 norm (see [9]) and therefore $\mathcal{H}(\mathbf{r})$ has a unique optimal solution and the approximation is thus well defined. Furthermore, the constraint set is comprised solely of linear constraints and hence constraint qualification criteria holds and the unique optimal solution is guaranteed to be the single set of values that satisfies the Karush-Kuhn-Tucker conditions [10]. Note that the gradient of the objective function, given by:

$$\left(\nabla\mathcal{H}(\mathbf{r})\right)_{ij} = -(1 + log(r_{ij})), \tag{3.12}$$

diverges to $-\infty$ as any of entries approach 0, and hence no strict non-negativity constraints are required as the optimal solution is guaranteed to be an inner point with $r_{ij} > 0$ for all $(i,j) \in E$. Let us define dual variables $V_1,\ldots,V_m$ and $W_1,\ldots,W_n$ for the customer and server flow constraints in Eq.(3.10) and Eq.(3.11) respectively. The KKT conditions imply that for the optimal solution, constraints (3.10),(3.11) are satisfied and, due to the stationarity condition, we must have

$$-(1 + log(r^*_{ij})) = V^*_i + W^*_j. \tag{3.13}$$

The constraints in (3.10)(3.11) are equality constraints and hence the dual variables can assume any sign. As a result, it easy to verify that we may rewrite stationarity condition

of(3.13) in the more illustrative form of

$$r_{ij}^* = \alpha_i \beta_j \cdot e^{(\tilde{V}_i^* - \tilde{W}_j^*)}. \tag{3.14}$$

Therefore, the maximum entropy approximation amounts to finding sets $V_1, \ldots, V_m$ and $W_1, \ldots, W_n$ that satisfy the following $m+n$ equations:

$$\sum_{j \in \partial(i)} r_{ij} = \sum_{j \in \partial(i)} \alpha_i \beta_j \cdot e^{(V_i - W_j)} = \alpha_i, \quad \textit{for } i = 1, \ldots, m \tag{3.15}$$

$$\sum_{i \in \partial(j)} r_{ij} = \sum_{i \in \partial(j)} \alpha_i \beta_j \cdot e^{(V_i - W_j)} = \beta_j, \quad \textit{for } j = 1, \ldots, n \tag{3.16}$$

which are in essence identical to the conditions of the Quasi-Independent Approximation suggested in [14]. Note that although the values of r_{ij} must be unique due the 1-strong concavity of the objective function (3.9), the dual variables lie a one-dimensional subspace of $\mathbb{R}^{m+n}$ since adding a constant to all dual variables does not change the resulting r_{ij} values. The reason for this is that (3.9) is a non-smooth function and hence the dual function is not strictly convex and the optimal solution is a k-dimensional subspace of the $m+n$ dimensional dual space with k being the number of connected components in the compatibility graph which ,due the assumption that CRP holds, is equal to 1. Although this a nonlinear problem it can be solved efficiently using the matrix scaling algorithm (Sheliekhovskii's method) which was shown by [11] to converge to the optimal solution. The method is also known in a more general form due to [43] as the Sinkhorn-Knopp iterations, described in Algorithm 1 the method was shown in [36] to have a linear convergence rate. The method has gained popularity in recent years as a method to approximate the optimal transport problem, and subsequently allow fast computation of Wasserstein distances [19]. For the case in question the method amounts to taking the $m \times n$ adjacency matrix A^E where

$$A_{ij}^E = \begin{cases} 1 & \textit{if } (i,j) \in E \\ 0 & \textit{otherwise}, \end{cases} \tag{3.17}$$

setting $A^0 = A^E$, and alternating between rescaling the rows so that the sum of the i'-th row equals α_i and rescaling the columns so that the sum of the j'-th column equals β_i at each iteration For a given ϵ we stop the algorithm if

$$\max \left(\max_{i=1,\ldots,m} \left(|\alpha_i - \sum_{j=1}^{n} A_{ij}^{\ell}| \right), \max_{j=1,\ldots,n} \left(|\beta_j - \sum_{i=1}^{m} A_{ij}^{\ell}| \right) \right) < \epsilon. \tag{3.18}$$

Note that at every iteration of the algorithm the *row scaling* operation is equivalent to a right multiplication of A^ℓ by a strictly positive diagonal matrix D_R^ℓ and every *column scaling* operation is equivalent to a left multiplication of $\tilde{A}^\ell$ by a strictly diagonal matrix D_L^ℓ. Hence, since the product of diagonal matrices is a diagonal matrix, at every iteration we have

$$A^\ell = \tilde{D}_L^\ell A^E \tilde{D}_R^\ell \tag{3.19}$$

Algorithm 1 Iterative Sinkhorn Projections

procedure SINKHORN(E, α, β)
  $\mathbf{A}^{(0)} \leftarrow A^E$, $k \leftarrow 0$
  while $max(||\alpha - A^{(\ell)}\mathbb{1}_m||_\infty, ||\beta^T - \mathbb{1}_n^T A^{(\ell)}||_\infty) > \epsilon$ **do**
    for $i = 1, \ldots, m$ **do**
      for $j = 1, \ldots, n$ **do**
        $\tilde{A}_{ij}^{\ell} = \alpha_i (\sum_{k=1}^{n} A_{ik}^{\ell})^{-1} A_{ij}^{\ell}$
      end for
    end for
    for $j = 1, \ldots, n$ **do**
      for $i = 1, \ldots, m$ **do**
        $A_{ij}^{\ell+1} = \beta_j (\sum_{k=1}^{m} A_{kj}^{\ell})^{-1} A_{ij}^{\ell}$
      end for
    end for
    $\ell \leftarrow \ell + 1$
  end while
  return $A^{(\ell)}$
end procedure

where $\tilde{D}_L^{\ell}$ and $\tilde{D}_R^{\ell}$ are both strictly positive diagonal matrices given by

$$\tilde{D}_L^{\ell} = D_L^{\ell} \cdot D_L^{\ell-1} \cdot \ldots D_L^{1}, \tag{3.20}$$

$$\tilde{D}_R^{\ell} = D_R^{1} \cdot D_R^{2} \cdot \ldots D_R^{\ell}. \tag{3.21}$$

Therefore, assuming the algorithm stops after N^ϵ iterations we set

$$V_i = \log\left((\tilde{D}_R^{N_\epsilon})_{ii}\right), \quad \textit{for } i = 1, \ldots, m \tag{3.22}$$

$$W_j = -\log\left((\tilde{D}_L^{N_\epsilon})_{jj}\right), \quad \textit{for } j = 1, \ldots, n \tag{3.23}$$

and retrieve an ϵ-approximate solution to (3.15) and (3.16). In practice we have found that a simple code implementation of the method provided results with precision of $\epsilon = 10^{-9}$ at a time scale of milliseconds, even for matrices with over 10^5 non-zero entries The resulting matrix provides us with the maximum entropy approximation by setting $r_{ij} = A_{ij}^{\ell}$. As we will demonstrate in subsequent sections, these values provide a remarkably accurate approximation of the FCFS infinite sequence matching rates.

Exact Cases

As a first base example we consider the completely connected compatibility graph with $E = \mathcal{I} \times \mathcal{J}$. For the completely connected graph no customers are ever passed by a server,

as the servers are qualified to serve all customers and hence the probability of a server j which appears with frequency of β_j to match with customer of type i is given by α_i and hence the matching rates are given by

$$r_{ij} = \alpha_i \beta_j, \quad \textit{for } (i,j) \in \mathcal{I} \times \mathcal{J}. \tag{3.24}$$

Note that if we set $V_i = c$ for $i = 1, \ldots, m$ and $W_j = c$ for $j = 1, \ldots, n$ the approximated values of r_{ij} in eq. (3.14) are equal to the exact matching rates, and since these satisfy the conditions of eq.(3.15)(3.16) they must also be the unique optimal solution of (3.9),(3.10),(3.11). Therefore, in the case of a completely connected compatibility graph the ME approximation will converge towards the exact matching rates.

Another somewhat trivial example is the case where the bipartite compatibility graph is a connected tree. A connected bipartite tree compatibility graph contains no loops and hence must have exactly $m + n - 1$ edges and will always have at least two leaf nodes, where a leaf node indicates either a customer class that may only be matched with a single server class or vice versa. If the compatibility graph is a connected bipartite tree, there is a single uniqeu solution to the equations (3.10),(3.11). To see this, consider the matching rate over an edge connecting a leaf node and observe that the matching rate over that edge is predetermined by the frequency of the leaf node. Therefore, the leaf node may be removed from the graph and the frequency of a neighbour of the leaf node can be reduced by the frequency of the leaf node without changing the feasible set of matching rates defined by constraints (3.10),(3.11). Now observe that after removing a leaf node from a connected bipartite graph the remaining graph is still a connected bipartite tree graph and hence must contain at least two leaves. This process can be repeated until all edges are eliminated. Therefore, the matrix scaling algorithm used to derive the approximation, which converges to a feasible solution, will converge to the exact matching rates.

Deriving the Maximum Entropy Approximation

Entropy-based approximations of steady state probabilities constrained on moment data have been shown to yield accurate stationary distribution approximations for a wide range of queueing systems, [37], [25]. However, in this section we do not apply the maximum entropy principle to the steady state probabilities, but rather we use it to derive matching rates based on the so called Gravity Model commonly used in transportation analysis, or more precisely on the justification underlying the use of the Gravity model in [56], [57]. In [14] the authors derive the same approximation of (3.15) and (3.16) using an analogous model with pipelines as the compatibility graph edges and pressures exerted at the nodes. We now demonstrate how the same approximation may be obtained through the use of a trip matrix approximation. This should not be surprising, as both the methods for computing flows of non-Newtonian fluid and the Gravity Model for trip matrix estimation are derived from the same principles of statistical mechanics. We now present the trip-matrix analogy as we will use it in a following section as our basis to derive an approximation for the matching rates of

an SBPSS. A common problem that arises in urban transportation analysis is the following: There are known to be N travelers from a set of origins $\mathcal{I} = \{1, \ldots, m\}$ to a set of destinations $\mathcal{J} = \{1, \ldots, n\}$. Given the count of travelers leaving each origin $o_1, \ldots, o_m$, and the count of travelers arriving at each destination $d_1, \ldots, d_n$ such that $N = \sum_{i=1}^{m} o_i = \sum_{j=1}^{n} d_j$, one wishes to estimate the entries of a trip matrix $T \in \{T \in \mathbb{N}^{m \times n} | \mathbb{1}^T T \mathbb{1} = N\}$ indicating the number of trips made between every origin-destination pair. The statistical mechanics based approach of [56] is to regard each individual traveler $1, \ldots, N$ as a distinct particle and every origin-destination pair $(i, j) \in \mathcal{I} \times \mathcal{J}$ as a possible state of the particle. The *micro-state* of the system is given by an N-tuple $\omega \in \mathcal{I} \times \mathcal{J}^N$ of origin-destination pairs describing the origin and destination of every traveler and the *macro-state* of the system is a matrix $T(\omega) \in \{T \in \mathbb{N}^{m \times n} | \mathbb{1}^T T \mathbb{1} = N\}$ which counts the number of particles found in each state. A micro-state $\omega \in \mathcal{I} \times \mathcal{J}^N$ is said to be feasible if both $T(\omega)\mathbb{1}_{m \times 1} = (o_1, \ldots, o_m)^T$ and $\mathbb{1}_{n \times 1}^T T(\omega) = (d_1 \ldots, d_n)$. The set of feasible micro-states is denoted by Ω and the set of feasible macro states is thus given by $\mathcal{T} = \{T(\omega), \omega \in \Omega\}$. The maximum entropy principle is applied by assuming that every distinct feasible micro-state $\omega \in \Omega$ is equally likely to occur and hence the likelihood of a given feasible macro-state is directly proportional to the number of micro-states that result in it, which we refer to as the multiplicity of the macro-state. In order to account for the impact of distance and/or travel times, a constraint of the form

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} T_{ij}(\omega) \leq C \tag{3.25}$$

is often imposed on the set of feasible micro states, where c_{ij} is proportional to the distance or travel time between origin i and destination j. In the context of an infinite matching sequence we wish to apply the same principle to derive an approximation of the infinite sequence matching rates. Given the customer and sever class frequencies and the compatibility graph, we construct a series of finite trip matrix estimation analogs of the matching sequence, replacing the origins, destinations and travelers with customers, servers and matches respectively. For $k \in \mathbb{N}$ let $o_{i,k} = \lfloor k\alpha_i \rfloor$ be the number of customers of class i for $i = 1, \ldots, n$ and let the $d_{j,k} = \lceil k\beta_j \rceil$ be the number of servers of type j for $j = 1, \ldots, n$. In the transportation setting of [56] it is a given that the number of travelers arriving at a destination must equal the sum of all trips to it, however due to rounding operations this may not hold for our purpose. To overcome this, we add another customer class numbered 0 with

$$o_{0,k} = \sum_{j=1}^{n} d_{j,k} - \sum_{i=1}^{m} o_{i,k} \tag{3.26}$$

and allow customer class 0 to match with any server type so that $E_0 = E \cup \{0\} \times \mathcal{J}$. Let us denote by

$$N_k = \sum_{i=0}^{m} o_{i,k} = \sum_{j=1}^{n} d_{j,k} \tag{3.27}$$

the total number of customer-server matches to be made. The constraint of (3.25) is replaced by a restriction of possible states of the particle to the set E_0 which is equivalent to enforcing constraint (3.25) with c_{ij} values given by

$$c_{ij} = \begin{cases} 0 & \text{if } (i,j) \in E_0 \\ C+1 & \text{if } (i,j) \notin E_0 \end{cases}. \tag{3.28}$$

The assumption of the CRP condition holding along with the fact that customer class 0 may match with any server, guarantee that the following equivalent conditions

$$\sum_{i \in I} o_{i,k} < \sum_{j \in \partial(I)} d_{j,k}, \quad \forall I \subsetneq \mathcal{I} \tag{3.29}$$

$$\sum_{j \in J} d_{j,k} < \sum_{i \in \partial(J)} o_{i,k}, \quad \forall J \subsetneq \mathcal{J} \tag{3.30}$$

$$\tag{3.31}$$

are satisfied, and hence by Hall's Theorem a perfect matching of the N_k customers and N_k servers is guaranteed to exist. The set of feasible micro states for the k-th system is given by

$$\Omega_k = \{\omega \in E_0^{N_k} | T(\omega)\mathbb{1}_{m+1} = (o_{0,k}, \ldots, o_{m,k})^T, \mathbb{1}_n^T T(\omega) = (d_{1,k}, \ldots, d_{n,k})\}. \tag{3.32}$$

The maximum entropy principle is applied and we assume any perfect matching $\omega \in \Omega_k$ has an equal probability $|\Omega_k|^{-1}$ of occurring. The probability of a macro state is thus directly proportional to its multiplicity given by:

$$w_k(T) = \frac{N_k!}{\prod_{(i,j) \in E_0} T_{ij}!}, \quad k \in \mathbb{N}. \tag{3.33}$$

For any pair of rational matrices $\mathbf{r}, \mathbf{r}' \in \mathbb{Q}^{(m+1)\times n}$ there exists an $N \in \mathbb{N}$ such that both $N \cdot \mathbf{r}, N \cdot \mathbf{r}' \in \mathbb{N}^{(m+1)\times n}$. Hence, as we are interested in the limiting case we may rewrite (3.33) as

$$w_k(\mathbf{r}) = \frac{N_k!}{\prod_{(i,j) \in E_0} (r_{ij} N_k)!}, \quad k \in \mathbb{N}. \tag{3.34}$$

where it is assumed $\mathbf{r} \in \mathbb{Q}^{(m+1)\times n}$. The most probable macro-state can be derived by maximizing any monotonically strictly increasing function of $w_k(\mathbf{r})$. Hence we define

$$\mathcal{H}_k(\mathbf{r}) = \frac{1}{N_k} \log \left(\frac{N_k!}{\prod_{(i,j) \in E_0} (r_{ij} N_k)!} \right), \quad k \in \mathbb{N}. \tag{3.35}$$

Replacing the log-factorials with Stirling's asymptotic approximation

$$\log(n!) \sim n \log(n) - n + \log \sqrt{2\pi n} + O\left(\frac{1}{n}\right), \tag{3.36}$$

we find that, in the limit, $\mathcal{H}_k(\mathbf{r})$ tends to a finite constant value independent of k:

$$\lim_{k\to\infty} \mathcal{H}_k(\mathbf{r}) \to -\sum_{(i,j)\in E} r_{ij}\log(r_{ij}) - \sum_{j=1}^{n} r_{0j}\log(r_{0j}) \tag{3.37}$$

Recall that by definition $o_{0,k} \leq m+n$ is independent of k, and hence as $r_{0j} \to 0$ and as $k \to \infty$ for all $j = 1, \ldots n$, the right-most term on the right hand side of (3.37) vanishes and we get $\mathcal{H}_k(\mathbf{r}) \to \mathcal{H}(\mathbf{r})$ as $k \to \infty$. Furthermore,

$$\frac{o_{i,k}}{N_k} \to \alpha_i \quad \textit{for } i = 1, \ldots, m \tag{3.38}$$

$$\frac{d_{j,k}}{N_k} \to \beta_j, \quad \textit{for } j = 1, \ldots, n \tag{3.39}$$

imply that enforcing $T \in \lim_{k\to\infty} \mathcal{T}$ can be achieved by enforcing constraints (3.10)(3.11) on the values $r_{ij}, (i,j) \in E$. Therefore, we have shown that for a trip-matrix analog of the matching-sequence, as the size of the system grows large the set of matching rates that will yield the most likely trip matrix can be derived by solving the constrained maximum entropy problem in (3.9),(3.10), (3.11). The fact that T^* is the most probable macro state does not necessarily imply anything about the probability of T^* occurring; however in this case as the size of the system grows, the maximum of $w_k(\cdot)$ becomes extremely sharp. Let $r_{i,j}, (i,j) \in E_0$ and $r'_{i,j}, (i,j) \in E_0$ be sets of matching rates that are realizable in $k-$th system such that $H_k((r)) > H_k((r'))$. The ratio of their multiplicity is given by:

$$\frac{w_k(\mathbf{r})}{w_k(\mathbf{r}')} = exp\{N_k(H_k(\mathbf{r}) - H_k(\mathbf{r}'))\} \tag{3.40}$$

and hence, as the system grows large the ratio of the probability of the most likely macro state to any other state with a strictly lesser value of $H(\cdot)$ passes any bound. In conclusion, our approximation in (3.15) and (3.16) is merely the result of assuming that for any finite length matching sequence, all matches are equally probable and then taking the limit of the length of sequence.

A Surprising Counter Example

Given the fundamental nature of the assumptions made in the previous subsection and given the relative accuracy of the approximation as will demonstrated in next section, one might expect the approximation scheme in (3.15)(3.16) to provide an accurate or approximately accurate calculation of the matching rates, with errors being the result of either limited accuracy of the convex optimization in (3.9), (3.10), and (3.11), or of a second order factor in the matching rate calculation that may vanish as the system grows large. Unfortunately, as we now show, this is not always the case. The example given is a surprisingly simple and counter-intuitive system. Let us define an (n,k) homogeneous chain, which we will denote by

$HC(n,k)$, for a given $n \geq k$, as a system with n customer classes $i = 0,\ldots,n-1$ and n server classes $j = 0,\ldots,n-1$, each with an equal frequency of $1/n$. The compatibility graph is a k-chain such that a customer of class i can match with server j only if $j \in \{mod_n(i+h)|h = 0,1,;k-1\}$. Let us now consider the $HC(5,3)$ in Figure 3.1, and to simplify the numbers let us re-scale the frequencies to be $3/4n$ instead of $1/n$ so that all a_i's and β_j's are set at 0.15. It easy to see that both our approximation of (3.15) and (3.16) and the approximations of

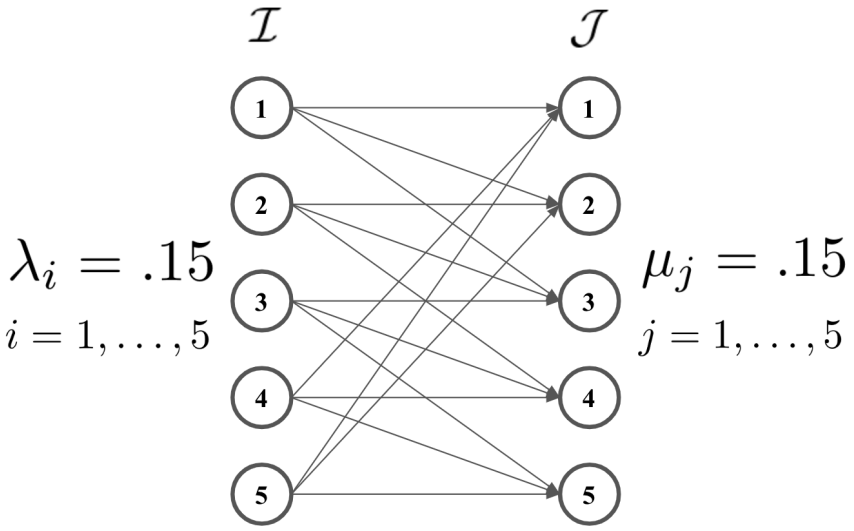


Figure 3.1: An $HC(5,3)$ system

[20] and [4] in (3.43),(3.44) will approximate the matching rate at $r_{i,j} = 1/15$ for each of the 15 arcs. Given the symmetry of the system one might expect the approximated values to agree with the exact values. This is not the case. The exact matching rates given by the Adan-Weiss closed form matching rates (3.6) show that $r_{0,0} = r_{0,2} = 17/250$ while $r_{0,1} = 16/250$, and the same pattern repeats for all customer nodes. Although at first glance it may seem that the three arcs of every node are indistinguishable, once we delve into the terms in (3.6) the asymmetry becomes apparent. The following explanation is based on the derivation of (3.6) from the beautiful state space formulation given in [3]. We now provide only some basic concepts that are required for the explanation, the interested reader may refer to [3] for an in-depth description of the state space. The explanation is based on the server-side description of the matching process described earlier in the section. The nodes of $HC(n,k)$ show a rotational symmetry: For any state of the system $(\mathfrak{s},n,\mathbf{v})$ there exist $n-1$ symmetric states that are equivalent to $(\mathfrak{s},n,\mathbf{v})$ in the sense that they are indistinguishable from a mere shift in the numbering of server and customer classes by $i \to mod_n(i+c)$ for $c = 1,\ldots,n-1$. In the $HC(n,k)$ systems all customer classes are served by exactly k servers and hence customers may only be waiting in slots $k, k+1,\ldots,n$. If any customer is waiting in slot k then it

may only be of a single customer class, the class that is only served by the first k servers. If a customer is in the k-th slot the probability of it matching to any specific compatible server class is identical across its compatible server classes as it is merely the probability of that server class being the first of the customer's compatible server classes to appear in the remaining server sequence. The probability that the next match occurring in the n-th slot (i.e., the next server in the sequence does not find a match after scanning all customers in the first $n-1$ slots) is dependent on the specific state of the system. However, given the symmetry of the states of the $HC(n,k)$ system, conditioned on having the match occurring at the the $n-th$ slot, the probability of any one of the possible $n\dot{k}$ matches occurring must be equal across all such possible matches. Hence, if we now focus on the example of the $H(5,3)$ system we can now conclude that any difference in the matching rates between the three arcs of every node must be a result of matches occurring in slot 4. Without loss of generality let us consider customer class 2 and focus on states where a customer of class 2 might match at the 4-th slot. For a match of a customer of class 2 to appear in the 4-th slot the system must be in a state where the first 4 servers are either in the subset $\{1,2,3,4\}$ or $\{2,3,4,5\}$. However, by symmetry of the $HC(5,3)$ system any state in which the servers $\{1,2,3,4\}$ appear first has a parallel state in which servers $\{2,3,4,5\}$ appear first, and hence we can focus on the first subset $\{1,2,3,4\}$. Let us condition on having the first 4 servers in the subset $\{1,2,3,4\}$ and observe that the only customer classes that may be found in the 4-th slot are classes 1 and 2. Let us consider the three possible cases:

1. The first three servers are either all in the subset $\{1,2,4\}$ or all in the subset $\{1,3,4\}$.
2. The first three servers are all in the subset $\{1,2,3\}$.
3. The first three servers are all in the subset $\{2,3,4\}$.

In the first case there can not be any customers in the 3-rd slot, as there is no customer class served only by the first three servers, and hence a 3-rd slot match may not occur and the probability of matching a customer of class 2 in the 4-th slot is equal for servers $2,3,4$. The second and third cases are, by symmetry, both equally likely and in both a 3-rd slot match may occur. The probabilities of matching a server of class $2,3$ to a customer in the 3-rd slot are equal because due to the rotation symmetry both cases 2 and 3 are equally likely and the probability of matching a server of class 1 to a customer in the 3-rd slot in the second case is equal to the probability of matching server of class 4 to a customer in the 3-rd slot in the third case. Hence, conditioned on having the first 4 servers being in the set $\{1,2,3,4\}$, servers 2 and 3 have an equal probability of not matching in the 3-rd slot and the servers 1 and 4 have an equal probability of not matching in the 3-rd slot. Furthermore, there is a higher probability of a 3-rd slot match, and hence a lower probability of a 4-th slot match, if the next server to appear is of either classes 2 or 3, as these may match on the 3-rd slot in both the second and third cases while servers 1 and 4 may match only in the second and third case respectively. The next server to appear is equally likely to be of any server class. If the next server to appear is either server 1 or 5 no match of a class 2 customer can

occur on the 4-th slot. If the next server to appear is either $2, 3$ the probabilities of it not matching to a customer in the 3-rd slot are equal and so are the probabilities of it matching to a customer of class 2 in the $4-$th slot given that they did not match on the 3-rd slot. Finally if the next server to appear is server 4, it has a higher probability than servers 2 and 3 of not matching on the 3-rd slot, and given that it did not match on the 3-rd slot it also has a higher probability of matching with a customer of class 2, as it can not match with customers of type 1.

We can now see that if the first four servers are in the subset $\{1, 2, 3, 4\}$, matches on the arc $(2, 4)$ are more likely than matches on the arcs $(2, 2)$ and $(2, 3)$. The rotational symmetry of $HC(5, 3)$ guarantees that the probabilities of matches on the arcs $(2, 3), (2, 3), (2, 4)$ in case the first four servers are in the set $\{1, 2, 3, 4\}$ are equal to the probabilities of matches on the arcs $(2, 4), (2, 3), (2, 2)$ respectively in case the first four servers are in the set $\{2, 3, 4, 5\}$. As we argued before, in case the first four servers are taken from any other subset, customers of class 2 may only match at the $5-$th slot where the sum of probabilities of matching on all arcs are equal. We conclude that the rates of matching on links $(2, 2)$ and $(2, 4)$ are equal and greater than the rate of matching on link $(2, 3)$. Therefore, we have found the "needle in the haystack," the asymmetry in a seemingly very symmetric system. In Figure 3.2, we consider $HC(n, k)$ systems and plot the difference between asymmetric matching rates of the simulation and the uniform matching rates predicted by the approximations both as function of n(left) and as a function of k(right). The graphs in Figure 3.2 provide some interesting

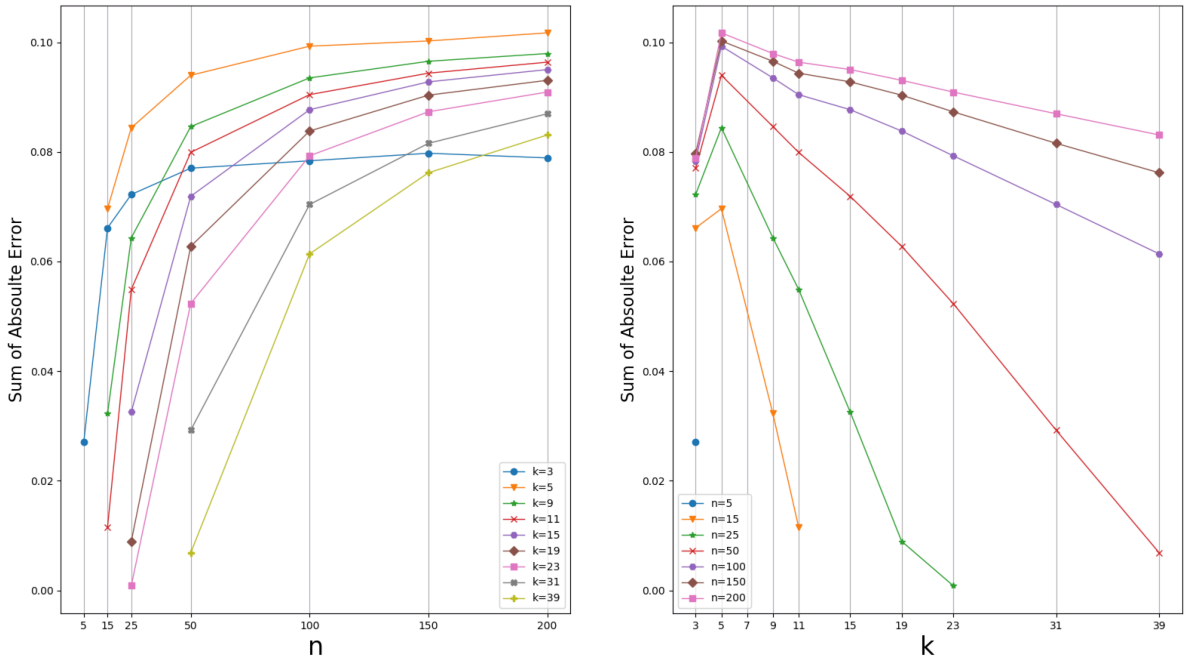


Figure 3.2: Sum of Absolute Approximation Error for the HC(n, k)

insights; the first being that for any length of chain the asymmetry grows as the size of the system grows. A second insight is that it seems that for a fixed size n the error is increasing up to $k = 5$

Comparison with Ohm's Law and Quadratic Approximations

In [20] the authors suggest an approximation scheme inspired by Ohm's Law. The bipartite compatibility graph is modeled as an electronic circuit with the customer class nodes and server class nodes as voltage sources and the edges as links. The customer class nodes and server class nodes are assigned voltages $V_1, \dots V_m$ and $W_1, \dots, W_n$, respectively, and the resistance of each link $(i, j) \in E$ is set to $(\alpha_i \beta_j)^{-1}$. The matching rate r_{ij} is approximated by the current on link (i, j) which by Ohm's Law is

$$I = \frac{\Delta V}{R}, \tag{3.41}$$

and thus the matching rate approximation is given by

$$r_{ij} = \alpha_i \beta_j \left(V_i - W_j\right) \quad \textit{for } (i, j) \in E \tag{3.42}$$

where the values V_i's and W_j's are determined by $m + n$ balance equations that enforce the sum of currents out of every customer node i to equal α_i and the sum of currents into every server node j to equal β_j:

$$\sum_{j \in \partial(i)} \alpha_i \beta_j (V_i - W_j) = \alpha_i, \quad i = 1, 2 \dots, m \tag{3.43}$$

$$\sum_{i \in \partial(j)} \alpha_i \beta_j (V_i - W_j) = \beta_j, \quad j = 1, 2 \dots, n. \tag{3.44}$$

The authors note that there is a single redundancy in the equations as $\sum_{i=1}^{m} \alpha_i = \sum_{j=1}^{n} \beta_j$; they suggest to arbitrarily set $W_n = 0$ as a default and remove one of the equations. The approximated matching rate on link i, j only depends on the differences between V_i and W_j and hence the choice of value for W_n and the removal of an equation leaves the approximation unchanged. Approximation (3.43)(3.44) can be easily obtained by solving a set of linear equations. Experiments in [20] show this approximation method provides reasonably accurate predictions of the matching rates for small Erdos-Renyi type graphs. However, a significant drawback of the approximation is that it may, as demonstrated in [4], in some cases predict negative matching rates. As a solution to this problem [4] suggest approximat-

ing the heavy traffic matching rates as the solution of the following quadratic program:

$$\min \mathcal{Q}(\mathbf{r}) = \sum_{(i,j)\in E} \frac{r_{ij}^2}{\alpha_i \beta_j} \tag{3.45}$$

subject to:

$$\sum_{j\in\partial(i)} r_{ij} = \alpha_i, \quad for\ i = 1, \ldots, m \tag{3.46}$$

$$\sum_{i\in\partial(j)} r_{ij} = \beta_j, \quad for\ j = 1, \ldots, n \tag{3.47}$$

$$r_{ij} \geq 0, \quad for\ i, j \in E. \tag{3.48}$$

[4] go on to show that, applying the KKT conditions, the unique optimal solution can be obtained by finding a set of values $\{V_1, \ldots V_m, \}, \{W_1, \ldots, W_n\}$ such that:

$$\sum_{j\in\partial(i)} \alpha_i \beta_j (V_i - W_j)^+, \quad i = 1, 2 \ldots, m \tag{3.49}$$

$$\sum_{j\in\partial(j)} \alpha_i \beta_j (V_i - W_j)^+, \quad i = 1, 2 \ldots, n, \tag{3.50}$$

where $x^+ = \max\{X, 0\}$. It is easy to see that in case the solution of the set of linear equations in (3.43), (3.44) is positive than the same solution would also satisfy the constraints of (3.49), (3.50), hence the quadratic program approximation of [4] can be seen as an enhancement of the Ohm's Law approximation that is guaranteed to provide a feasible (and positive) set of matching rates at the cost of requiring the solution of a quadratic rather than a linear program. The approximation scheme suggested by [20] and [4] resemble the approximation of (3.15) and (3.16) in that there is a single value V_i for each customer node $i = 1, \ldots, n$ and a single value W_j for every server node $j = 1, \ldots n$ and the approximation of every matching rate r_{ij} is given by multiplying the complete graph matching rate $\alpha_i \beta_j$ by a function of V_i and W_j. In the case of the Ohm's Law Approximation the function is $V_i - W_j$, but in our Maximum Entropy Approximation the function is $\exp(V_i - W_j)$. In both cases the values of the V_i's and W_j's are determined by enforcing the flow balance constraints.

In the remainder of this section we repeat the experiments in section 3.3 of [20] and compare the accuracy of the Ohm's Law Approximation (3.43), (3.44), the Quadratic Program Approximation of ((3.46) ,(3.47), and our maximum entropy approximation as in (3.16) ,(3.15). The simulation experiment in section 3.3 of [20] considers a random bipartite graphs with $m = 10$ customer classes and n server classes where n is uniformly distributed between 7 and 10. Each one of the possible $m \cdot n$ edges is either included or not included in the compatibility graph with a probability of 50%. Any graph where two customer (server) class are linked to the exact same set of servers (customers) is rejected as the two are indistinguishable and can be collapsed into a single class. The graphs are classified into three

types based on the density of the edges: *High Density* $|E|/m \cdot n \geq 70\%$, *Medium Density* $40\% < |E|/m \cdot n < 70\%$ and *Low Density* $|E|/m \cdot n \leq 40\%$, and 15 graphs of each type are generated. The customer class frequencies $\alpha_1, \ldots, \alpha_m$ and server frequencies are generated. Both customer and server frequency vectors are generated by drawing m and n values from a random distribution and than scaling the values so that they sum up to unity. For every graph 20 pairs of sets $\alpha_1, \ldots, \alpha_m$ and $\beta_1, \ldots, \beta_m$ are generated using an exponential distribution, and 20 are generated using a uniform distribution. In total, 1800 experiments (3 *densities* $\times$15 *graphs* $\times$40 *frequencies*) are performed. To fairly compare the results of approximations, we compute the exact matching rates using equation (3.6) from [3] and compare performance of the approximations to the exact solutions. In order to perform the computations we developed a scheme for efficient computation of (3.6) using the Steinhaus–Johnson–Trotter algorithm [32] to allow us to iterate over all permutations by only replacing pairs of adjacent servers at every iteration and thus considerably reducing the number of value retrievals required[1]. In [20] the authors use Mean Absolute Error Rate (MAE) (avg. across edges of the absolute approximation error) to estimate the accuracy of the Ohm's Law approximation. The use of MAE as a measure of accuracy can be misleading as the mean rate per edge itself depends on the graph density. Therefore, the same MAE value can be the result of either a poor approximation of the matching rates in a dense graph, where the actual match rate on each individual edge is small, or an accurate approximation of the matching rates in a sparse graph where the individual matching rates are higher. We provide the MAE in Table 3.2 only to demonstrate that our estimates of the accuracy of the Ohm's Law approximation are of the same order of magnitude as those in [20]. However, the authors [20] do not specify their treatment of the negative values that may arise in the approximation and we speculate that this is the cause of the differences as the sum of errors we calculated by excluding all negative rates (SNR) is closer to the values reported in [20]. The approximation accuracy measures we wish to compare are the sum of absolute errors (SAE) and the Maximum Absolute Error (MXAE) between the approximated and exact matching rate on a single edge. The SAE is equivalent to the portion of the matching rate that must be reassigned in order for the approximation to equal the exact matching rates. The SAE is merely a rescaling of the MAE for any specific instance, however, as all matching rates sum to 1 on all instances it allows a comparison of approximation accuracy across systems with a varying number of edges. Table 3.1 reports the averages of both the sum and the maximum of the absolute differences between the exact and approximated matching rates.

As shown in [4] the Ohm's Law approximation produces the same results as the Quadratic approximations whenever the first provides positive approximations, hence we need only compare the performance of the Max. Entropy approximation with that of the Quadratic approximation. As may be seen, in 3.1 the Maximum Entropy Approximation provides a lower avg sum of absolute error for all graph densities. While for the high density graphs the performance gap is minor, for the medium and lower density graphs the Maximum Entropy Approximation is substantially more accurate than the Quadratic Approximation. If we

[1] we refer the reader to `https://github.com/dgrosbar/FSS` for code and implementation details

Density	Approximation	Avg. SAE	Max. SAE	Avg. MAXE	Avg. MAE
High	*Ohm's Law*	.0329	.2049	.0029	$.495\times10^{-3}$
	Quadratic	.0314	.1235	**.0027**	$.472\times10^{-3}$
	Max Entropy	**.0296**	**.058**	.0031	$.441\times10^{-3}$
Medium	*Ohm's Law*	.0896	.3463	.0095	1.989×10^{-3}
	Quadratic	.0631	.1424	.0064	1.390×10^{-3}
	Max Entropy	**.0345**	**.0727**	**.0033**	0.756×10^{-3}
Low	*Ohm's Law*	.1145	.4319	.0132	3.266×10^{-3}
	Quadratic	.0648	.1370	.0071	1.834×10^{-3}
	Max Entropy	**.028**	**.064**	**.0029**	0.792×10^{-3}

Table 3.1: Infinite FCFS Matching Sequence Match Rate Approximation Errors

Density	Avg. MAE	Avg. MAE in [20]	Avg. SNR
High	$.495\times10^{-3}$	$.2\times10^{-3}$	$.5\times10^{-3}$
Medium	1.989×10^{-3}	$.8\times10^{-3}$	7.778×10^{-3}
Low	3.266×10^{-3}	1.6×10^{-3}	12.613×10^{-3}

Table 3.2: MAE comparison with results of[20]

consider the worst result under all densities we can see that in the worst case the Quadratic approximation mis-allocates 14.24% of the flow, while the worst case error of Maximum Entropy Approximation never exceeds 7.2% as shown in Table 3.1. If we count the instances where one approximation outperforms the other, we can see that the Quadratic Approximation outperforms the Maximum Entropy Approximation for 61% of the high density graphs while for the medium and low density graphs the Maximum Entropy Approximation outperforms the Quadratic approximation 83.1% and 95.5% of the time, respectively. Note that even for the high density graphs, where the Quadratic approximation outperformed the Max. Entropy approximation on 61% of the cases it did so by an Avg. margin of .0102 and a maximum margin of 0.0344 while for the 39% of cases where the Max Entropy approximation outperformed the Quadratic approximation it did so by an Avg. margin of .0207 and considerable Max. margin of .1035. In general, table 3.3 shows that for any graph density, in cases where the Quadratic approximation outperformed the Maximum Entropy Approximation it does so by a small margin while in cases where the Max. Entropy Approximation outperforms the Quadratic Approximation it often does so by a substantial margin. Next we compare the approximations on large scale SBPSSs of two types, a bipartite Erdős-Rényi graph with 1000 nodes on each side and a *2-Tours* graph with 900 customer class nodes and

Density	Lower SABSE	No. Cases	Avg. Margin	Max. Margin
High	*Quadratic*	**366(61%)**	.0102	.0344
	Max Entropy	234(39%)	**.0207**	**.1035**
Medium	*Quadratic*	103(17.1%)	.0065	.0245
	Max Entropy	**497(82.9%)**	**.0359**	**.1081**
Low	*Quadratic*	30(5%)	.007	.0223
	Max Entropy	**570(95%)**	**.0389**	**.1146**

Table 3.3: Comparison of the Quadratic and Maximum Entropy approximations across cases

Structure	Approximation	Avg. SAE	Max. SAE	Min. SAE
Erdős-Rényi 1000-1000	*Ohm's Law*	.131	.221	.081
	Quadratic	.085	.0966	.068
	Max Entropy	**.0588**	**.0672**	**.0508**
2-Torus (30x30)-(30x30)	*Ohm's Law*	.109	.145	.0892
	Quadratic	.109	.145	.0892
	Max Entropy	**.0538**	**.0546**	**.0523**

Table 3.4: Infinite FCFS Matching Sequence Approximation SAE for large scale SBPSSs

900 server nodes laid out on a 30x30 Torus grid (see Figure. 1.5) with edges connecting any pair of nodes that are no further than two grid lines apart. For each graph structure we created 30 random instances and simulated 30 repetitions for every instance with each repetition consisting of 10^7 customer arrivals. The results in Table 3.4 show that there is a slight degradation in the average accuracy of the entropy approximation going from and avg. SAE of 2.5-3% in the small cases to an SAE of 5.5% larger graphs. However, it also appears that the Max. Entropy approximation remains robust for large scale graphs with maximum errors of 6.7% and 5.4% on the instances of Erdős-Rényi and 2-Torus graphs respectively, furthermore the extremely small span of error for the *2-Tours* suggests that the accuracy entropy approximation depends more on the topology of the graph , which is identical across all *2-Torus* instances. Finally we can see that while for small denser graphs the Quadratic approximation could outperform the entropy approximation for some instances, in the case of the larger graphs the Max. Entropy approximation dominates as for both structures the Min. SAE achieved by the QP approximation $(6.8\%, 8.9\%)$ was larger than the Max. SAE obtained by the Entropy approximation (6.72%, 5.46%).

3.2 Fluid Approximation of the Infinite ALIS Sequence Matching Rates

In an ALIS stochastic matching model we are given a random infinite sequences of customers and a random initial permutation of the servers. The customers and servers are of classes $\mathcal{I} = \{1, \ldots, m\}$ and $\mathcal{J} = \{1', \ldots, n'\}$ respectively and a bipartite graph $G = (\mathcal{I} \cup \mathcal{J}, E)$ defines the compatible customer-server pairs. The class of every customer $i^k, k \in \mathbb{N}$ is an $i.i.d$ random variable that assumes the value $i \in \mathcal{I}$ with probability α_i where $||\boldsymbol{\alpha}||_1 = 1$. The ALIS matching procedure works as follows: At the k-th match instance, customer $i^k \in \mathcal{I}$ in the customer sequence scans the server permutation starting from the first position until it finds the first compatible server $j \in \partial(i)$ and matches with the server, the match is logged as an (i, j) match and the customer is removed from the sequence. After the match the server assumes the last position in the permutation and all servers that were behind the matched server move one position forward. An illustration of the process is given in Figure 3.3. Let

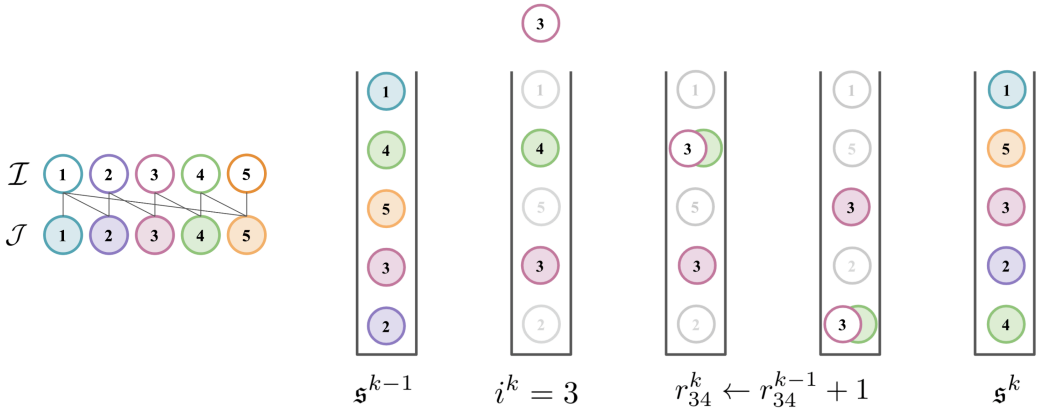


Figure 3.3: The Matching Process in an ALIS Matching Sequence

r^k_{ij} be the count of (i, j) matches after a total of k matches have been made. The matching rates of the infinite ALIS matching sequence are defined as:

$$r_{ij} = \lim_{k \to \infty} \frac{r^k_{ij}}{k}. \tag{3.51}$$

As was shown in [2] under the a heavy traffic regime the matching rates of an SBPSS with server dependent service time operating under a FCFS-ALIS policy converge to the matching rates of the FCFS infinite matching sequence. We now wish to show that under a light traffic regime the matching rates of of an SBPSS with server dependent service time operating under a FCFS-ALIS policy converge to the matching rates of the ALIS infinite matching sequence.

The Light Traffic SBPSS

Consider a series of systems with server dependent service time where $\mu_j^{(N)} = N \cdot \mu_J$ operating under an LQF-ALIS or FCFS-ALIS policy. In such a system as $N \to \infty$ the matching rates and service rates become independent. To see this, consider the steady state probabilities from [2]. For a state where the first k servers are busy and ordered by the arrival time of the customers they are currently serving and the integer v_k is the count of customers that have been skipped by servers $k+1, \ldots, J$ due to incompatibility. Then

$$\lim_{N\to\infty} \pi(\mathfrak{s}, k, \mathbf{v}) = \lim_{N\to\infty} B \prod_{\ell=1}^{k} \frac{\lambda_{(\ell)}^{n_\ell}}{(N\mu_{(\ell)})^{n_\ell+1}} \prod_{\ell=k+1}^{n} \bar{\lambda}_{(\ell)}^{-1}. \tag{3.52}$$

For a given permutation of the severs $\mathfrak{s} \in \mathfrak{P}_{\mathcal{J}}$ and a integer $k \in \{0, \ldots, n\}$, the steady state probability of finding the system in a state at which the servers are ordered by permutation $\mathfrak{s}$ and the first k servers being busy is given by:

$$\lim_{M\to\infty} P(\mathfrak{s}, k) = \lim_{N\to\infty} \sum_{n_1,\ldots,n_k} \pi(\mathfrak{s}, k, \mathbf{v}) = \lim_{N\to\infty} B \prod_{\ell=1}^{k} (N \cdot \mu_{(\ell)} - \lambda_{(\ell)})^{-1} \prod_{\ell=k+1}^{n} \bar{\lambda}_{\ell}^{-1} \tag{3.53}$$

and we have that for any $k = 1, \ldots, n$,

$$\lim_{N\to\infty} \frac{P(\mathfrak{s}, k)}{P(\mathfrak{s}, 0)} = \lim_{N\to\infty} \frac{\prod_{\ell=k+1}^{n} \bar{\lambda}_{(\ell)}}{\prod_{\ell=1}^{k} (N \cdot \mu_{(\ell)} - \lambda_{(\ell)})} = 0. \tag{3.54}$$

Therefore, the proportion of arrivals that find a system with busy servers vanishes as $N \to \infty$ and the matching rates are determined only by the stationary probabilities of those state where all servers are idle. The stationary probability of a state where all servers are idle depends only on the arrival rates and the normalizing constant B. However under light traffic we have

$$B = \lim_{N\to\infty} \sum_{\mathfrak{s}\in\mathfrak{P}_{\mathcal{J}}} \sum_{k=0}^{n} P(\mathfrak{s}, k) = \lim_{N\to\infty} \sum_{\mathfrak{s}\in\mathfrak{P}_{\mathcal{J}}} P(\mathfrak{s}, 0) \sum_{k=0}^{n} \frac{P(\mathfrak{s}, k)}{P(\mathfrak{s}, 0)} = \sum_{\mathfrak{s}\in\mathfrak{P}_{\mathcal{J}}} P(\mathfrak{s}, 0) \tag{3.55}$$

and hence B itself only depends on the arrival rates but not the service rates. In this case the FCFS priority among the different queues does not play a rule in the decision as the probability of an event where two customers of different types are waiting simultaneously for the same server vanishes. As a result, the matching rates in a system under light traffic will converge to the matching rates of the corresponding infinite ALIS matching sequence. The matching process under a light traffic regime can be described as follows:

- The servers are stacked in a column with the upper server being the one that has been idle for the longest time period and the bottom one being the last server to go idle.

- Upon arrival, a customer will pass through the servers in the stack from the top down until it encounters the first compatible server, at which point the customer class will match with the server.
- Under a light traffic regime we assume that customer will always be served and leave the system before the next customer appears. Hence, we regard the service as instantaneous and the matched server immediately sinks down to the bottom of the stack.
- all servers below the previous position of the matched server move up one slot to fill in the vacancy left by the matched server.

The ALIS Markov Chain

The states of the system under light traffic are restricted to the set $\{\pi_{\mathfrak{s},0,0_n} | \mathfrak{s} \in \mathfrak{P}_{\mathcal{J}}\}$ and can therefore be represented solely by the permutation $\mathfrak{s} \in \mathfrak{P}_{\mathcal{J}}$. Given a system $\mathcal{F}$ in light traffic let $X_v \in \mathfrak{P}_{\mathcal{J}}, v \in \mathbb{N}_0$ be a discrete Markov chain with states representing the ordering of the stack with X_0 being any initial state and X_v describing the state of the system after v customer arrivals. We state the following Theorem:

Theorem 3.2.1. *Let $\mathcal{F} = (\mathcal{I} \cup \mathcal{J}, E, \boldsymbol{\lambda}, \mathbf{s}, \boldsymbol{\mu})$ be such that for any $j, j' \in \mathcal{J}, \partial(j) \subseteq \partial(j') \rightarrow j = j'$ then the discrete Markov chain $X_v \in \mathfrak{P}_{\mathcal{J}}, v \in \mathbb{N}_0$ is Ergodic*

Proof. The adjacent permutation graph of the set $\mathcal{J}$ denoted by $PG(\mathcal{J}) = (\mathfrak{P}_{\mathcal{J}}, E_{adj})$ consists of a set of $n!$ nodes representing the permutations of $\mathcal{J}$ and an edge set E_{adj} such that $(\mathfrak{s}, \mathfrak{s}') \in E_{adj}$ if and only if $\mathfrak{s}$ and $\mathfrak{s}'$ differ from one another by a single swap of adjacent values. For example, in the graph $PG(\{1, 2, 3\})$ we have that $((1, 2, 3), (2, 1, 3)) \in E_{adj}$ as one can swap the adjacent positions of elements 1 and 2 to obtain one from the other, in contrast $((1, 2, 3), (3, 2, 1)) \notin E$ as obtaining one from the other requires swapping the positions of elements 1 and 3 which are not adjacent. The proof of the Theorem is based on the fact that $PG(\mathcal{J})$ is known to contain a Hamiltonian cycle (see [32]) and therefore, if we can show that given a state $\mathfrak{s}$ one can reach any state $\mathfrak{s}'$ such that $(\mathfrak{s}, \mathfrak{s}') \in E_{adj}$ within a finite number of arrivals and with a strictly positive probability than the irreducibility of the Markov chain will follow immediately. First, let us observe that, as we assume every server $j \in \mathcal{J}$ serves some customer class there is always a strictly positive probability that the server at the top of the stack will be matched with the next arriving customer. Let us consider the permutation describing the state of the system as a ring with the server at top of the stack to the right of the server at the bottom the stack as illustrated in Figure 3.4. A match at the top of the stack will cause each element in the ring to shift one position to the left. Furthermore, note that although a matching at the first server changes the permutation it does not change the relative positions of the servers. Let us refer to a series of $k \in \mathbb{N}$ consecutive matches of the customer at the top of the stack as a k-rotation. Let $\mathfrak{s}, \mathfrak{s}' \in \mathfrak{P}_{\mathcal{J}}$ be states of the system such that $(\mathfrak{s}, \mathfrak{s}') \in E_{adj}$ and $\mathfrak{s}'$ can be obtained from $\mathfrak{s}$ by swapping the serves j_k, j_{k+1} at positions $k, k + 1$. We will now describe a finite sequence of arrivals that may occur with a strictly positive probability and will cause the Markov chain to transition from state $\mathfrak{s}$ to state $\mathfrak{s}'$:

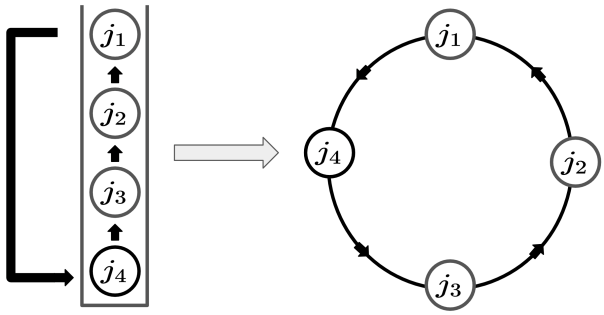


Figure 3.4: The State of the System as a Ring

1. A $v-1$ rotation will have the servers j_{k+1} and j_k at positions $2, 1$ respectively
2. An arrival of a customer class $i \in \partial(j_{k+1}) \setminus \partial(j_k)$ will send server $k+1$ to position n and leave server j_k at position 1. Note that by assumption $\partial(j_{k+1}) \setminus \partial(j_k) \neq \emptyset$.
3. An $n-k$ rotation will have servers j_{k+1} and j_k at positions k and $k+1$ respectively.

Note that after each arrival in the sequence, any server other than j_{k+1}, j_k moved exactly one position to the left and since there were a total of $(k-1)+1+(n-k) = n$ arrivals each server other than j_{k+1}, j_k has returned to its initial position while j_k, j_{k+1} have swapped places. Hence, any state of the Markov chain is reachable from any other state and the Markov chain is irreducible. Given any state of the system an $n-$rotation will results in the system returning to the same state. Furthermore, since $\partial(j) \subseteq \partial(j') \to j = j'$ we know that at any state there is a strictly positive probability of the next customer arrival to be of a class that is compatible with the server at position 2 but incompatible with the server at position 1 and hence it easy to verify that after $n-1$ such arrivals the system will return to the same state. Any state of the Markov chain can thus be returned to with either n or $n-1$ transitions and hence the Markov chain is aperiodic. A finite state irreducible, aperiodic Markov chain is Ergodic. □

An Ergodic Markov chain admits a stationary distribution over the state of the chain. For any permutation matrix $\mathfrak{s} \in \mathfrak{P}_{\mathcal{J}}$ let $\pi_{\mathfrak{s}}$ be the stationary probability of state $\mathfrak{s}$. An $n \times n$ permutation matrix is a matrix with a single 1 entry on every column and every row and 0 at all other entries. The set of $n \times n$ permutation matrices has the following one-to-one correspondence with the set of permutations (see Section 1.1 of [12]): A permutation $\mathfrak{s} \in \mathfrak{P}_{\mathcal{J}}$ that has servers $1, 2, 3, \ldots, n$ at positions $j_1, j_2, \ldots, j_n$ respectively corresponds to a permutation matrix that has 1 entries at $(1, j_1), (2, j_2), \ldots, (n, j_n)$ and 0 elsewhere. For any permutation $\mathfrak{s} \in \mathfrak{P}_{\mathcal{J}}$ we denote by $\bar{\mathfrak{s}}$ the correspoding permutation matrix. Let us define

$$\mathbf{p}^{(\pi)} = \sum_{\mathfrak{s} \in \mathfrak{P}_{\mathcal{J}}} \pi_{\mathfrak{s}} \bar{\mathfrak{s}} \tag{3.56}$$

and observe that in steady state, the probability that a random arrival finds server j at the k-th position is given by:

$$Pr\{server\ j\ \ in\ position\ k\} = \sum_{\substack{\mathfrak{s}\in\mathcal{P}_n \\ \bar{\mathfrak{s}}_{jk}=1}} \pi_{\mathfrak{s}} = \mathbf{p}^{(\pi)}_{jk} \tag{3.57}$$

Furthermore, the matrix $P^{(\pi)}$ is a convex sum of permutation matrices and hence by the *Birkhoff-von Neumann* theorem it must also be a doubly stochastic matrix. Our approximation of the ALIS matching rates is based on an approximation of the matrix $P^{(\pi)}$ using an analogous fluid system.

The Fluid ALIS model

In order to approximate $\mathbf{p}^{(\pi)}$ we first replace the single server stack with N stacks and scale the arrival rate so that $\boldsymbol{\lambda}^{(N)} = N \cdot \boldsymbol{\lambda}$, we refer to this system as the parallel stack system. In the parallel system, an arriving customer will go down a given stack with an equal probability of $1/N$. This parallel-stack system itself is not yet an approximation as each identical column is equivalent to the original. However, we wish to make a few observations regarding the parallel-stack that will motivate the development of the final approximation scheme. Let $X^N_v \in \mathfrak{P}^N_{\mathcal{J}}, v \in \mathbb{N}_0$ be the discrete Markov chain whose states are N-tuples $(\mathfrak{s}_1, \ldots, \mathfrak{s}_N) \in \mathfrak{P}^N_{\mathcal{J}}$ of permutations specifying the order of each of the N stacks with X^N_0 being any initial state and X^N_v describing the state of the system after v customer arrivals. We refer to X^N_v as the micro-state of the system. Let $Y^N_v = \frac{1}{N}\sum_{\ell=1}^N \bar{\mathfrak{s}}_\ell$ be the macro-state of the system. First, note that at any micro-state of the parallel stack system there are exactly N servers that are at the k-th position of their stack and exactly N servers of each type $j \in \mathcal{J}$. As a result, at any micro-state, the proportion of servers of type j that are at the k-th position of their respective stack is equal to the proportion of the servers at the k-th layer of the stack that are of type j. Therefore, the macro-state Y^N_v is a doubly stochastic matrix. In addition, the long term proportion of time in which each individual stack is at a state $\mathfrak{s} \in \mathfrak{P}_{\mathcal{J}}$ is given by $\pi_{\mathfrak{s}}$ and hence $\pi_{\mathfrak{s}}$ is also the long term avg. proportion of stacks that are at state $\mathfrak{s}$ and therefore we have:

Proposition 3.2.1.

$$\lim_{\substack{N\to\infty \\ v\to\infty}} Y^N_v = \sum_{\mathfrak{s}\in\mathfrak{P}_{\mathcal{J}}} \pi_{\mathfrak{s}}\bar{\mathfrak{s}} = \mathbf{p}^{(\pi)}$$

Proposition 3.2.1 implies that an approximation of $\lim_{v,N\to\infty} Y^N_v$ is equivalent to an approximation of $\mathbf{p}^{\pi}$. Let us now replace the parallel stack system with a multistack system in order to obtain an approximation of $\lim_{v,N\to\infty} Y^N_v$. In the multistack system an incoming customer, instead of going down a specific stack, follows a random downward path across the stacks so that at any given layer $k-1, \ldots, n$ the customer may encounter the k-th server of any stack with probability $\frac{1}{N}$. An incoming customer will thus encounter a random server at

each layer until it encounters a compatible server at some layer $k = 1, \ldots, n$. If the customer encounters a compatible server at the n-th layer, the system remains unchanged, otherwise, after encountering a compatible server at layer $k < n$, the matched server will sink to the bottom n-th layer and a random server will be displaced and move up from layer n to $n-1$ displacing another random server that will move up from layer $n-1$ to layer $n-2$ and so forth. The displacement process will continue until a server from layer $k+1$ moves up to layer k and fills the vacant slot caused by the match. In the multistack process there is no guarantee that an incoming customer will indeed encounter a compatible server as it passes through the n layers. Therefore, a customer that does not encounter a compatible server at any of the n-layers will immediately repeat the process again starting from the first layer until it is eventually matched. Let $X_v^N \in \{1, \ldots, n\}^{n \times N}, v \in \mathbb{N}_0$ be the discrete Markov chain whose states are $n \times N$ matrices describing the server type at each location of the multistack with X_0^N being the initial state and X_v^N describing the state of the system after v customer arrivals. Let $\hat{Y}_v^N \in \{[0,1]^{n \times n}|, v \in \mathbb{N}_0\}$ be an $n \times n$ matrix such that the entry at the k-th row and j-th column of $\hat{Y}_v^N$ denotes the portion of k-th layer servers that are of type j after v customer arrivals. Note that, as in the parallel-stack system, there are still N servers of every type $j \in \mathcal{J}$ and N servers per stack layer k. For this reason, the portion of k-th layer servers that are of type j is again equal to the portion of type j servers that are at the k-th layer and hence $\hat{Y}_v^N$ is also a doubly stochastic matrix. Given a multi-stack system at some initial macro-state $\hat{Y}_0^N = \mathbf{p}$ the probability of a customer of class $i \in \mathcal{I}$ arriving at the first layer, possibly after having gone unmatched through all n layers before, to go unmatched at the first $k-1$ layers and arrive at the $k = 1, \ldots, n$ layer is given by:

$$q_{ik}(\mathbf{p}) = \prod_{\ell=1}^{k-1}(1 - \sum_{j \in \partial(i)} \mathbf{p}_{\ell,j}) \tag{3.58}$$

The number of times an arriving customer will have to pass through the stack unmatched is a geometric random variable with mean given by

$$C_i(\mathbf{p}) = \left(1 - \prod_{k=1}^{n}(1 - \sum_{j \in \partial(i)} \mathbf{p}_{k,j})\right)^{-1} \tag{3.59}$$

If the state of system is held fixed (for example a system where matched servers do not sink), then the rate at which a customer of type $i \in \mathcal{I}$ matches with a server of type $j \in \partial(i)$ at layer k is given by

$$r_{ij}^k(\mathbf{p}) = \mathbf{p}_{kj} \cdot q_{ik}(\mathbf{p}) C_i(\mathbf{p}) \lambda_i, \quad \textit{for } (i,j) \in E, k = 1, \ldots, n \tag{3.60}$$

Let us denote the rates of all server j matches at layers $\ell = 1, \ldots, k$

$$M_j^k(\mathbf{p}) = \sum_{\ell=1}^{k} \sum_{i \in \partial(j)} r_{ij}^\ell(\mathbf{p}) \tag{3.61}$$

and denote the rate of all server matches layers $\ell = 1, \ldots, k$ by

$$M^k(\mathbf{p}) = \sum_{j \in \mathcal{J}}^{k} M_j^k(\mathbf{p}) \tag{3.62}$$

Let us now consider the fluid limit of the multistack system as $N \to \infty$. The discrete arrivals are replaced with a constant volumetric flow rate λ_i and the macro state of the discrete Markov chain Y_v^N is replaced with the continues time dynamic system $\mathbf{p}(t) \in [0,1]^{n \times n}$ where $\mathbf{p}_{kj}(t)$ denotes the portion k-th layer servers that are of type j. To abbreviate notation we let $\mathbf{p}(t) = \mathbf{p}$. Figure 3.5 illustrates the transition from the server stack ALIS model to the approximation using a stack of fluids. The dynamics of $\mathbf{p}$ are given by the following set of

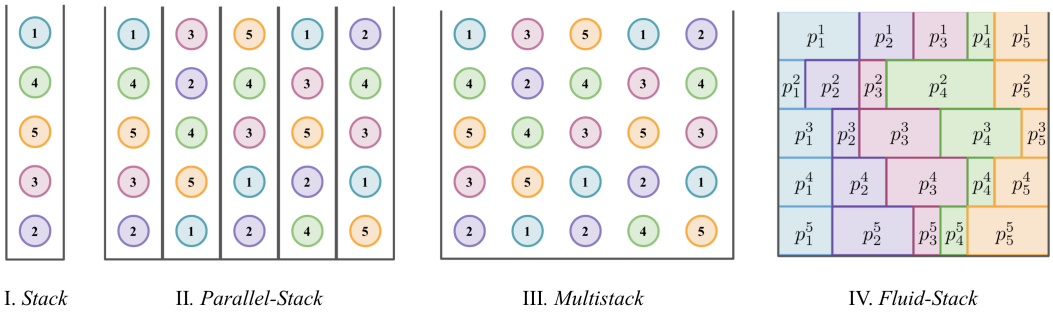


Figure 3.5: Transition from a Stack model to the Fluid Stack model

ordinary differential equations

$$\begin{aligned}
\dot{\mathbf{p}}_{1j} &= \mathbf{p}_{j2} M^1(\mathbf{p}) - \sum_{i \in \partial(j)} r_{ij}^1(\mathbf{p}) && , j \in \mathcal{J} \\
\dot{\mathbf{p}}_{kj} &= \mathbf{p}_{k+1,j} M^k(\mathbf{p}) - \sum_{i \in \partial(j)} r_{ij}^k(\mathbf{p}) - \mathbf{p}_{kj} M^{k-1}(\mathbf{p}) && , j \in \mathcal{J}, k = 2, \ldots, n-1 \\
\dot{\mathbf{p}}_{nj} &= \sum_{\ell=1}^{n-1} \sum_{i \in \partial(j)} r_{ij}^{\ell}(\mathbf{p}) - \mathbf{p}_{nj} M^{n-1}(\mathbf{p}) && , j \in \mathcal{J}
\end{aligned} \tag{3.63}$$

Proposition 3.2.2. *If* $\mathbf{p}(0)$ *is doubly stochastic then* $\mathbf{p}(t)$ *is doubly stochastic for any* $t \geq 0$

Proof. If $\mathbf{p}$ is doubly stochastic than for any $k = 1, \ldots, n$ we have that $\sum_{j \in \mathcal{J}} \mathbf{p}_{kj} = 1$ and

hence

$$\sum_{j\in\mathcal{J}} \dot{\mathbf{p}}_{kj} = \sum_{j\in\mathcal{J}} \mathbf{p}_{k+1,j} M^k(\mathbf{p}) - \sum_{j\in\mathcal{J}} \sum_{i\in\partial(j)} r_{ij}^k(\mathbf{p}) - \sum_{j\in\mathcal{J}} \mathbf{p}_{kj} M^{k-1}(\mathbf{p}) =$$
$$M^k(\mathbf{p}) \sum_{j\in\mathcal{J}} p_{k+1,j} - M^{k-1}(\mathbf{p}) \sum_{j\in\mathcal{J}} p_{k+1,j} - \sum_{j\in\mathcal{J}} \sum_{i\in\partial(j)} r_{ij}^k(\mathbf{p}) =$$
$$M^k(\mathbf{p}) - M^{k-1}(\mathbf{p}) - \sum_{j\in\mathcal{J}} \sum_{i\in\partial(j)} r_{ij}^k(\mathbf{p}) = 0$$

and it is easy to verify that $\sum_{k=1}^{n} \dot{\mathbf{p}}_{kj} = 0$ for any $j \in J$ as the terms of the form $p_{kj} M^k(\mathbf{p})$ cancel each other out when summing across the first $k-1$ equations and the terms of the form $\sum_i \in \partial(j) r_{ij}(\mathbf{p})$ cancel out with the summation at the LHS of the final equation. Therefore, if $\mathbf{p}(0)$ is doubly stochastic it will remain so as the initial row and column sums of $\mathbf{p}(t)$ remain unchanged. □

These fluid equations capture the transitions of the multistack system. For a layer $k = 1, \ldots, n-1$ in the multistack system a server is randomly pushed in from the $k+1$ layer every time a match occurs at a layer $\ell \leq k$ and a random server is pushed out of layer k every time a match occurs at a layer $\ell < k$. Therefore, the rate at which servers of type j are pushed into a layer k is the product of the sum of matching rates at layers $\ell \leq k$ given by $M^k(\mathbf{p})$ and $p_{k+1,j}$ the proportion of layer $k+1$ servers that are of type j and similarly, the rate at which servers are pushed out of a layer k is the product of the sum of matching rates at layers $\ell < k$ given by $M^{k-1}(\mathbf{p})$ and the proportion of layer k servers that are of type j given by p_{kj}. The rate at which servers of type j leave layer k is the sum of the rates at which they match at layer k and the rate at which they are pushed out of layer k. The rate at which type j servers enter layer k is the rate at which they are pushed in from the $k+1$-th layer. At the $n-th$ layer the servers enter by sinking and hence the rate of entry of j type servers is the sum of the matching rates of j type servers at layers $1, \ldots, n-1$. A fluid state $\mathbf{p}$ is stable if $\dot{\mathbf{p}} = 0$ which from 3.63 implies that for any layer $k = 1, \ldots, n$ and for any server type $j \in J$ the rate at which servers of type j leave layer k is equal to rate at which they enter. Setting $\dot{\mathbf{p}} = 0$ in (3.63) yields the following stationarity conditions:

$$\mathbf{p}_{1j} = 1 - \sum_{k=2}^{n} p_{kj} \qquad , j \in \mathcal{J} \tag{3.64}$$

$$\mathbf{p}_{kj} = \frac{M_j^{k-1}(\mathbf{p})}{M^{k-1}(\mathbf{p})} \qquad , j \in \mathcal{J}, k = 2, \ldots, n \tag{3.65}$$

Fixed Point Iteration

A general solution procedure for the set of differential equations in (3.63) is not known to us, nor is a process for the determination of the existence of a unique solution. Nonetheless, we have found that a simple fixed point iteration procedure performs remarkably well in practice.

Observe that if the values of $\mathbf{p}$ are held fixed the values of $r_{ij}^k(\mathbf{p})$ are strictly determined by (3.60). Furthermore, given a set of matching rate values $r_{ij}^k, (i,j) \in E, k = 1, \ldots, n$ there is a unique set of values, $\mathbf{p}(\mathbf{r})$ such that (3.65) holds for rows $k = 2, \ldots, n$ of $\mathbf{p}(\mathbf{r})$ and (3.60) holds for the first row, albeit with these values $\mathbf{p}(\mathbf{r})$ need not necessarily be doubly stochastic. At the same time, the Sinkhorn-Knopp Theorem states that, since $\mathbf{p}(\mathbf{r})$ is strictly positive, there is a unique $n \times n$ doubly stochastic matrix of the from $D_L \mathbf{p}(\mathbf{r}) D_R$ where D_L, D_R are $n \times n$ diagonal matrices[2]. Hence we propose a fixed point iteration scheme which alternates between deriving a set of matching rates $\mathbf{r}$ from a given proportion matrix $\mathbf{p}$ and deriving a row stochastic matrix $\mathbf{p}$ from a set of matching rates $\mathbf{r}$ and projecting it onto the space of doubly stochastic matrices via Sinkhorn-Knopp iterations. A fixed point of such a procedure would by definition be a doubly stochastic matrix for which (3.64), (3.65) hold and will thus be a stable point of the dynamic system in (3.63) which we denote by $\hat{\mathbf{p}}^{(\pi)}$ and will constitute our approximation of $\mathbf{p}^{(\pi)}$. Once $\mathbf{p}^{(\pi)}$ is approximated the corresponding matching rates are given by setting $\hat{\mathbf{p}}^{\pi}$ into (3.60) and summing the matching rates across the n layers. Let us define $f_{\mathbf{p}\to\mathbf{r}} : \mathbb{D}^{n\times n} \to \mathbb{R}_+^{m\times n\times n}$ be the function that, given a doubly stochastic matrix $\hat{\mathbf{p}} \in \mathbb{D}^{n\times n}$ returns a set of matching rates $\mathbf{r} \in \mathbb{R}_+^{m\times n\times n}$ using (3.60) and let $f_{\mathbf{r}\to\mathbf{p}} : \mathbb{R}_+^{m\times n\times n} \to \mathbb{R}_+^{n\times n}$ be the function that, given a set of matching rates $\mathbf{r} \in \mathbb{R}_+^{m\times n\times n}$ returns a matrix $\hat{\mathbf{p}} \in \mathbb{R}_+^{n\times n}$ by reversing (3.60) to obtain the first row of $\hat{\mathbf{p}}$ and using (3.65) to obtain rows $k = 2, \ldots, n$. The detailed light traffic approximation procedure is given in Algorithm 2 which uses the SINKHORN$(\mathbf{p}, \alpha, \beta)$ procedure from Algorithm 1 as a subroutine. The procedure in Algorithm 2 was found to converge for all cases in the

Algorithm 2 Light Traffic Approximation Procedure

procedure ALIS-APPROXIMATION(E, λ)
  $\mathbf{p}^{(k)} \leftarrow \frac{1}{n}\mathbb{1}_n\mathbb{1}^T$, $k \leftarrow 0$
  while $||\mathbf{p}^{(k)} - \mathbf{p}^{(k-1)}|| > \epsilon$ **do** ▷ Condition only checked after first iteration
    $\mathbf{r}^{(k)} \leftarrow f_{\mathbf{p}\to\mathbf{r}}(\mathbf{p}^{(k)})$
    $\hat{\mathbf{p}}^{(k+1)} \leftarrow \begin{bmatrix} \mathbf{p}_{11}^{(k)}, \ldots, \mathbf{p}_{1n}^{(k)} \\ f_{\mathbf{r}\to\mathbf{p}}(\mathbf{r}^{(k)}) \end{bmatrix}$
    $\mathbf{p}^{(k+1)} \leftarrow$ SINKHORN$(\hat{\mathbf{p}}_{k+1}, \mathbb{1}_n, \mathbb{1}_n)$
    $k \leftarrow k + 1$
  end while
  $\mathbf{r} \leftarrow f_{\mathbf{p}\to\mathbf{r}}(\mathbf{p}^k)$
  $\mathbf{r} = \sum_{\ell=1}^n \mathbf{r}_\ell$ ▷ $\mathbf{r}_\ell$ is the $m \times n$ matrix of matching rates at layer ℓ
  return r
end procedure

small scale experiments of Section 3.1 using both a uniform doubly stochastic matrix and a random strictly positive doubly stochastic matrix as initial values for $\mathbf{p}$. The algorithm

[2] D_L, D_R are themselves unique modulo multiplication of one and division of the other by a non-zero constant

also converged for the large scale Erdős-Rényi and Torus graphs of Section 1.5 however the algorithm could not converge for Map graph instances for which the CRP condition does not hold. This leads us to conjecture that the convergence of the fixed point iteration depends on the CRP condition.

Simulation Results - ALIS approximation

To test the accuracy of the ALIS approximation we repeat the experiment of [20] using the same randomly generated graphs as in section 3.1 and comparing the approximated matching rates with the results of simulations of the ALIS stochastic matching models. The results of the approximations are summarized in table 3.5 Hence we can see that the approximation

Density	Avg. SAE	Max. SAE	Avg. MAXE
High	.0142	.0401	.0011
Medium	.0432	.0786	.0049
Low	.0319	.0703	.0034

Table 3.5: ALIS Matching Rate Approximation Error Rates

performs well for all three graph densities with the lowest error rate of 1.4% for the high density graphs and the highest error rate of 4.32% on the medium density graphs. The worst case approximation over 1800 experiments was 7.86% which indicates that approximation is as robust as the equivalent approximations for the infinite FCFS bipartite matching sequence.

Structure	Avg. SAE	Max. SAE	Min. SAE
Erdős-Rényi 1000-1000	.0291	.0465	.021
Torus (30x30)-(30x30)	.0422	.0693	.0335

Table 3.6: Infinite ALIS Matching Sequence Approximation SAE for large scale SBPSSs

Next we tested the accuracy on the results on the large scale Erdős-Rényi and Torus graph using the same randomly generated instances as in Section 3.1 and validated that the approximations accuracy as can be seen in Table 3.6. The convergence of the fixed point iteration was relatively fast taking on avg. 2.78 seconds to converge for with a worst case of 4.02 seconds for Torus graph no.7 using a 2.3 GHz Intel Core i5 processor. Improved code efficiency may still potentially reduce the convergence time. To conclude, the aforementioned results confirm that we thus derived what is, to the best of our knowledge, the first approximation scheme for the matching rates of the infinite ALIS bipartite matching sequence.

3.3 Approximating the FCFS-ALIS Matching Rates of a Skill SBPSS

Having formulated approximations for the matching rates of the skill based parallel service system under both heavy-traffic and light-traffic regimes via the FCFS and ALIS infinite matching sequence approximations we now wish to provide a unified approximation scheme for the SBPSS under any traffic intensity. As can be expected, our approximation will consist of a convex combination of the heavy traffic approximation, based on the infinite FCFS matching sequence and a light traffic approximation based on the infinite ALIS matching sequence. Nevertheless, direct use of the asymptotic regime approximations may, as we will demonstrate yield poor approximations or in some cases non-admissible approximations and therefore, we first we present schemes to adjust the approximations individually for varying traffic intensities.

The FCFS Approximation

We now describe an approximation scheme suited for sub-critical SBPSSs at high traffic intensities. By "high traffic intensity" we are not referring to a heavy traffic regime with $\rho \to 1$, but rather to a systems where the probability of a customer finding an idle queue upon arrival is low, in which case the matching process is dominated by the FCFS component of the FCFS-ALIS policy. The proposed approximation of the matching rates in a sub-critical skill-based parallel service system is closely related to the matching rates of an infinite matching sequence. As demonstrated in [2] a parallel service system with service-dependent service times operating at critical loading, where the sum of all customer rates approaches the sum of all service rates, behaves as an infinite matching sequence and the matching rates converge to those given in [3]. The approximation we propose for the matching rates in a sub-critical skill based parallel service system with homogeneous service requires two adjustments to the Max Entropy approximation of (3.9), (3.10) ,(3.11). The first adjustment is to replace the direct approximation of matching rates $r_{ij}, (i,j) \in E$ with an approximation of the workload removal rates $\eta_{i,j}, (i,j)$, where $\eta_{i,j} = r_{ij}s_i$ is the rate at which a server $j \in \mathcal{J}$ removes workload introduced to the system by a customer class $i \in \partial(j)$. The set of admissible workload removal rates is homeomorphic to the set of admissible matching rates and is defined by

$$\Delta_{\eta,\mu} = \left\{ \eta \in \mathbb{R}^{(m+1)\times n} \;\middle|\; \sum_{j\in\partial(i)} \eta_{ij} = \eta_i, \forall i \in \mathcal{I} \text{ and } \sum_{i\in\partial(j)} \eta_{ij} = \mu_j, \forall j \in \mathcal{J} \right\} \tag{3.66}$$

The second adjustment is the addition of another *idleness* customer class that "*fills*" the gap between the sum of services rates and the sum of workload arrival rates. The *idleness* class, given the index 0, has an arrival and service rate of

$$\lambda_0 = \sum_{j\in\mathcal{J}} \mu_j - \sum_{i\in\mathcal{I}} \lambda_i s_i \quad , \quad s_0 = 1 \tag{3.67}$$

and we refer to the system with the additional customer class as the *compacted* system. The workload removal rates of the compacted system are approximated by the matching rates of an infinite matching sequence with frequencies corresponding to the workload arrival rates $\eta_i, i \in \mathcal{I}$ and service rates $\mu_j, j \in \mathcal{J}$. The matching rates of the compacted system provide an approximation of the workload removal rates in the sub-critical SBPSS. Hence, there are two levels of approximation, we first approximate the SBPSSs by an appropriate infinite FCFS bipartite matching sequence and we then continue and approximate the matching rates of that sequence. The approximated workload removal rates of a subcritical SBPSS with traffic intensity are thus given by the solution of the following maximum entropy problem:

$$\max \mathcal{H}_0(\mathbf{r}) + \mathcal{H}(\mathbf{r}) = -\sum_{(i,j)\in E} \eta_{ij} \log(\eta_{ij}) - \sum_{j\in\mathcal{J}} \eta_{0j} \log(\eta_{0j}) \tag{3.68}$$

subject to:

$$\sum_{j\in\partial(i)} \eta_{ij} = \eta_i, \qquad \textit{for } i = 1, \ldots, m \tag{3.69}$$

$$\sum_{i\in\partial(j)} \eta_{ij} + \eta_{0j} = \mu_j, \qquad \textit{for } j = 1, \ldots, n \tag{3.70}$$

which by applying KKT condition is equivalent to finding values $V_i, i = 0, \ldots, m$ and $W_j, j = 1 \ldots, n$ that satisfy the following equations:

$$\sum_{j\in\partial(i)} \eta_{ij} = \sum_{j\in\partial(i)} \eta_i \mu_j \cdot e^{(V_i - W_j)} = \eta_i, \quad \textit{for } i = 0, \ldots, m \tag{3.71}$$

$$\sum_{i\in\partial(j)} \eta_{ij} = \sum_{i\in\partial(j)} \eta_i \mu_j \cdot e^{(V_i - W_j)} = \mu_j, \quad \textit{for } j = 1, \ldots, n \tag{3.72}$$

Once, the workload removal rates have been approximated the corresponding matching rates are given by

$$r_{ij} = \frac{\eta_{ij}}{s_i} \tag{3.73}$$

The proposed approximation uses the same maximum entropy approach as the heavy traffic approximation. As such, one may consider using a rescaling of the heavy traffic approximation as an approximation of the matching rates when the traffic intensity is high. However, as we now wish to demonstrate, even at a very high traffic intensity the matching rates of a system can be considerably different from those obtained under a heavy traffic regime. Let us consider a series of skill-based parallel service systems indexed by $n \in \mathbb{N}$. The n-th system is comprised of a set customer classes $\mathcal{I} = \{1, \ldots, n\}$ each arriving at a rate of $\lambda_i = \rho/n$ for some $\rho \in (0, 1)$ and a set of servers $\mathcal{J} = \{1', \ldots, n'\}$ each serving any qualified customer class at a rate of $\mu_j = 1/n$. A customer of class $i \in \mathcal{I}$ is qualified to be served by all servers with a smaller or equal index so that $E = \{(i, j') | i \geq j\}$ is the set of qualified customer-class

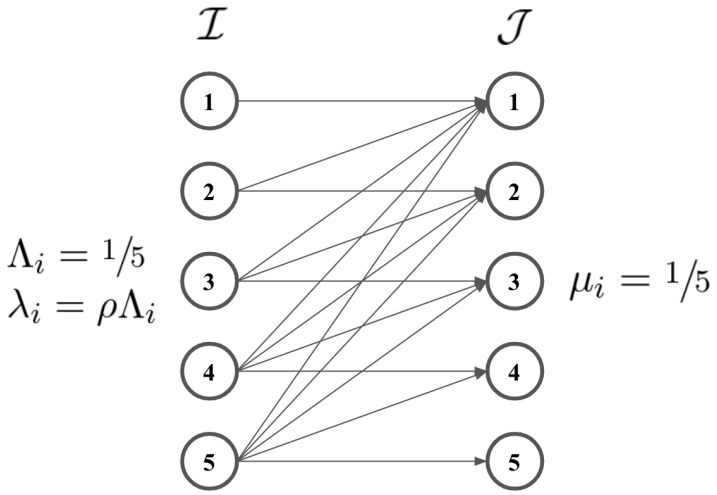


Figure 3.6: *Increasing-N* system for $n = 5$

to server assignments. We refer to this system as the *Increasing-N* system. An illustration of such a system for $n = 5$ is given in Figure 3.6. For the *increasing-N* system with $\rho \to 1$ every customer and server pair (i, i') form a distinct CRP component. In Section 4 of [4] the authors prove that the matching rates of arcs connecting different CRP components approaches 0 as $\rho \to 1$ and hence under a heavy traffic regime the portion of matches occurring on the arcs of $E_=$ approaches 1. However, results of simulations in Figure 3.7 show the heavy traffic regime would produce a poor estimation of the matching rates for a large sub-critical *increasing-N* system for any traffic intensity $\rho < 1$. For any $\epsilon > 0$ and $\rho < 1 - \epsilon$ the portion of the matching that occurs on the arcs of $E \setminus E_=$ increases with n; for example, with $\rho = .99$ and $n = 100$, over 30% of the matches occur on the edges of $E \setminus E_=$. The matching rates on $E_\neq$ vanish under the heavy traffic regime[3] but are very well-approximated by approximation (3.71), (3.72) which approximate the system with an SAE of less than 2% compared to simulation for all values of n that were tested.

The ALIS approximation

As established in Section 3.2 , the matching rates of an SBPSS under light traffic converge to the matching rates of an infinite ALIS matching sequence. A significant property of a system under light traffic regime is that as the traffic intensity approaches 0 the matching rates become independent of both the workload requirements $\{s_i | i \in \mathcal{I}\}$ and service rates $\{\mu_j | j \in \mathcal{J}\}$. Clearly, this property does not hold when the traffic intensity increases. In

[3] An interesting result of [4] is that while the matching rates vanish on $E_\neq$, the presence of $E_\neq$ still has a considerable impact on the scaling of waiting times

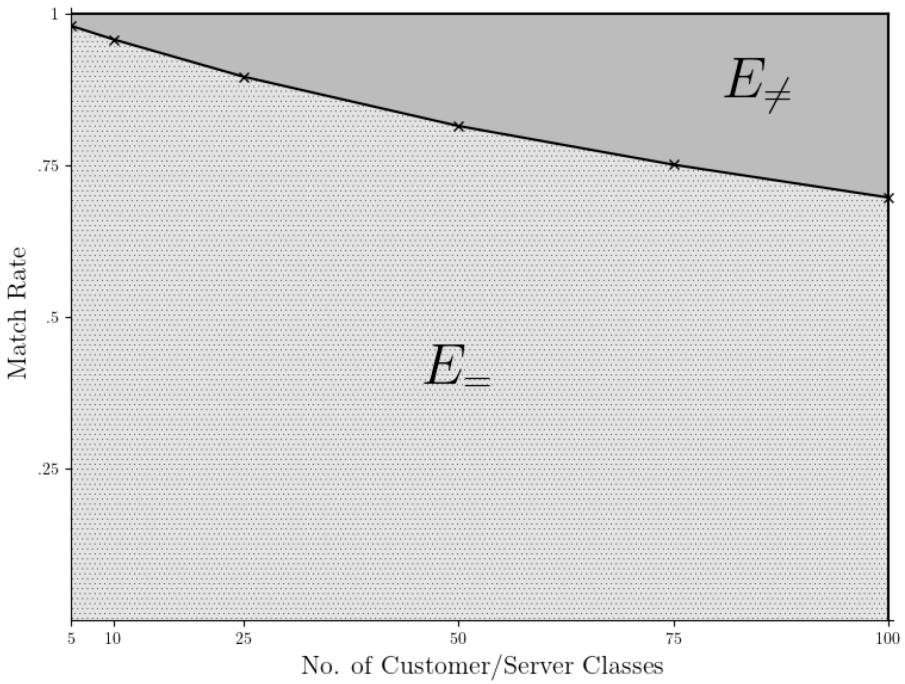


Figure 3.7: Matching rates on the edges of $E_{=}, E_{\neq}$ in an *increasing-N* system as a function of system size

order for the approximation to accommodate strictly positive traffic intensities we must alter the approximation. Instead of approximating the matching rates by a doubly stochastic proportion matrix that satisfies (3.63) we seek a row stochastic proportion matrix where the rows sum up to 1 as before but the columns sum to a vector $\hat{\boldsymbol{\mu}}(\rho)$ given by

$$\hat{\boldsymbol{\mu}}(\rho) = (1-\rho) + n\rho\boldsymbol{\mu} \tag{3.74}$$

so that $\hat{\boldsymbol{\mu}}(\rho) \to 0_n$ as $\rho \to 0$ and $\hat{\boldsymbol{\mu}}(\rho) \to \mathbb{1}_n$ as $\rho \to 1$. In practice this requires two adjustments to the fixed point iteration. First, we replace initial uniform proportion matrix $\mathbf{p}$ with the matrix $\mathbf{p}^{(0)} = \frac{1}{n}\mathbb{1}_N\tilde{\mu}^T$ with row sums of 1 and column sums $\hat{\mu}(\rho)$. Second, instead of projecting the set of proportions to the set of doubly stochastic matrices at every iteration we apply $\textsc{Sinkhorn}(\mathbf{p}^{(k)}, \mathbb{1}_n, \hat{\boldsymbol{\mu}})$ and project the set of proportions onto the set of matrices with row sums of 1 and column sums $\hat{\boldsymbol{\mu}}(\rho)$. In essence, if we go back to the derivation of the approximation illustrated in Figure 3.5 we are adding server particles for those servers with higher service rates and removing server particles for servers with lower service rates in an attempt to account for the impact of congestion.

The FCFS-ALIS approximation

For the light and heavy traffic regimes the ALIS and FCFS approximations coincide the respective infinite matching sequence approximations the accuracy of which has been established empirically. We now propose using a convex combination of the two approximations as an approximation for the workload removal rates of a subcritical SBPSS under the FCFS-ALIS. The weights given to the FCFS and ALIS approximations should reflect the ration of server side, FCFS matches and customer side, ALIS matches. A natural choice is thus to use the systems traffic intensity ρ as we know that for $\rho = 1$ all matches will be server side matches and for $\rho = 0$ we will only havw customer side matches. Let $\hat{\mathbf{r}}^{ALIS}(\rho), \hat{\mathbf{r}}^{FCFS}(\rho)$ denote the FCFS and ALIS match rate approximations for a given system at a traffic intensity of $\rho \in [0, 1]$ the proposed approximation is:

$$\hat{\mathbf{r}}(\rho) = \rho \cdot \hat{\mathbf{r}}^{FCFS}(\rho) + (1 - \rho) \cdot \hat{\mathbf{r}}^{ALIS}(\rho) \tag{3.75}$$

To the best of our knowledge there is no other approximation schemes for the matching rates of a sub critical SBPSS under a FCFS-ALIS policy. In order to estimate the accuracy of our approximation we repeat the experiment of [20] and of Sections 3.1, 3.2 and use the error rates of the heavy-traffic and light-traffic approximations of the FCFS and ALIS infinite matching sequences as benchmarks. Let $\alpha_i, i \in \mathcal{I}$ be the customer class frequencies in the infinite FCFS matching sequence experiments of Section 3.1. In order to test the approximation for subcritical systems we let $\eta_i = \alpha_i \times \rho$ for some $\rho \in (0, 1)$ so that the traffic intensity of the entire system is ρ. We introduce an a additional vector of parameters $\theta \in \mathbb{R}_+^m$ that distributes the workload of the customer class i between arrival rate and service requirement such that

$$s_i = \eta_i^{\theta_i} \quad \text{and} \quad \lambda_i = \eta_i^{(1-\theta_i)} \tag{3.76}$$

and consider four scenarios: Three scenarios where θ is uniform with $\theta_i = 0, \frac{1}{2}, 1$ and a scenario where $\theta_i \sim Uniform[.1, .9]$. Note that when $\theta = 0$ we have server dependent service times as in [2]. The graphs in Figures 3.8 -3.11 show the rSAE which is the ratio of SAE to the overall arrival, i.e the portion of the total arrivals that was misplaced by the approximation for different traffic intensities. Furthermore, as both the FCFS and ALIS approximation do not explicitly rely on the FCFS policy we compare the approximated matching rates to the simulated matching rates of the systems under both FCFS-ALIS and the LQF-ALIS policies. The results in Figures 3.8 -3.11 display three key contributions of this work. First and foremost, if the QP and Ohm's Law approximations of the infinite FCFS matching sequences are taken as benchmarks for accuracy, the FCFS-ALIS approximation is thus, to the best of our knowledge, the first valid approximations of SBPSS matching rates across all utilization levels. The rSAE for the FCFS-ALIS approximation is highest for traffic intensities in the range of $.3 - .4$ where the error is approximately 11% across all experiments which is still lower than the Avg. SAE of the Ohm's law approximation for small low density graphs. At higher traffic intensity levels of $.65 - .99$ the FCFS-ALIS reaches very low rSAE levels of approximately $1\% - 5\%$, this is an improvement over the accuracy of the infinite

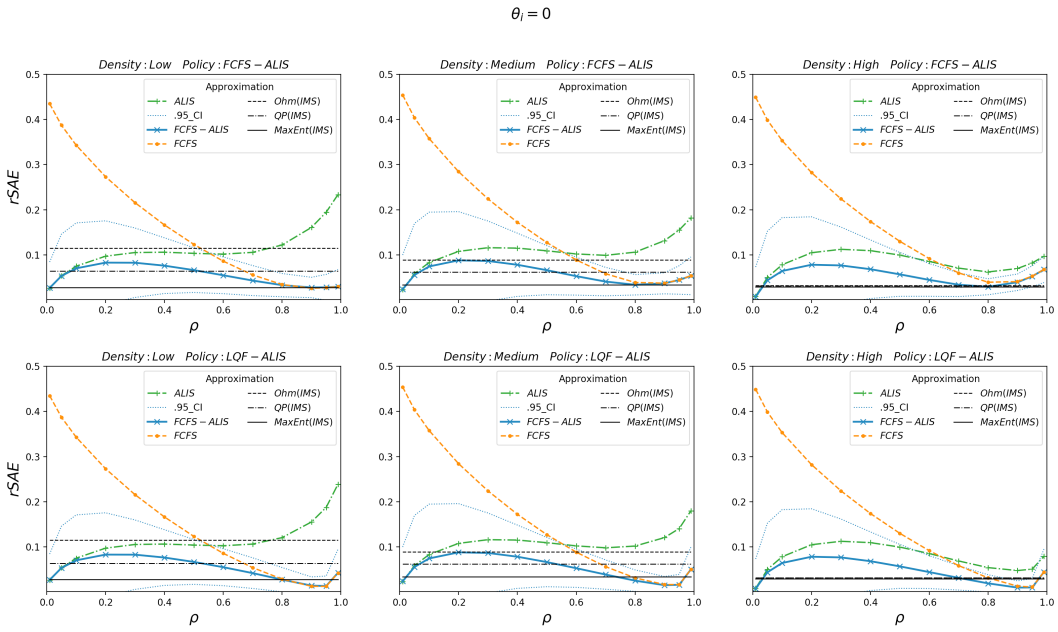


Figure 3.8: Error Rate by Traffic intensity for $\theta_i = 0, i \in \mathcal{I}$

matching sequence approximations and is also encouraging as this is the desired range of operation most applications. A second important contribution that arises from these results is the fact that the accuracy of the FCFS-ALIS approximation does not seem to depend at all on the workload distribution vector $\boldsymbol{\theta}$, this is an important observation as the exact matching rates of [2] are only derived for the case of server dependent service times ($\boldsymbol{\theta} = 0$) and hence the approximation enables us to analyze a wider range of systems.

Finally, the results clearly indicate that the FCFS-ALIS is in fact an FCFS/LQF-ALIS approximation. In some cases the approximations are closer to the LQF-ALIS matching rates than those of the system under an FCFS-ALIS policy. This further expands the applicability of the approximation. The fact that the matching rates under both the LQF-ALIS and FCFS-ALIS are similar should not be too surprising as the correlation between the waiting time of a customer in the queue and the length of the queue that accumulated behind it is obvious. Next we turn to larger scale graphs for which the calculation of exact matching rates, even for the heavy traffic case, are not tractable. The three graph structure considered are Bipartite Erdos-Renyi graphs with 1000 nodes on each side, a 30x30 tours grid graph with 900 customer class nodes, 900 server nodes and edge distance of 2 and a 30x30 map graph and a edge distance of 2. The results for both LQF-ALIS and FCFS-ALIS approximation appear in Figure 3.12 Experiments with map graphs indicate that convergence of ALIS approximation requires that CRP to hold. As such, ALIS approximations could not be obtained and we present only the results of the FCFS approximation. Nonetheless, despite a clear deprecation in the accuracy compared to the small instances, the FCFS-ALIS appears

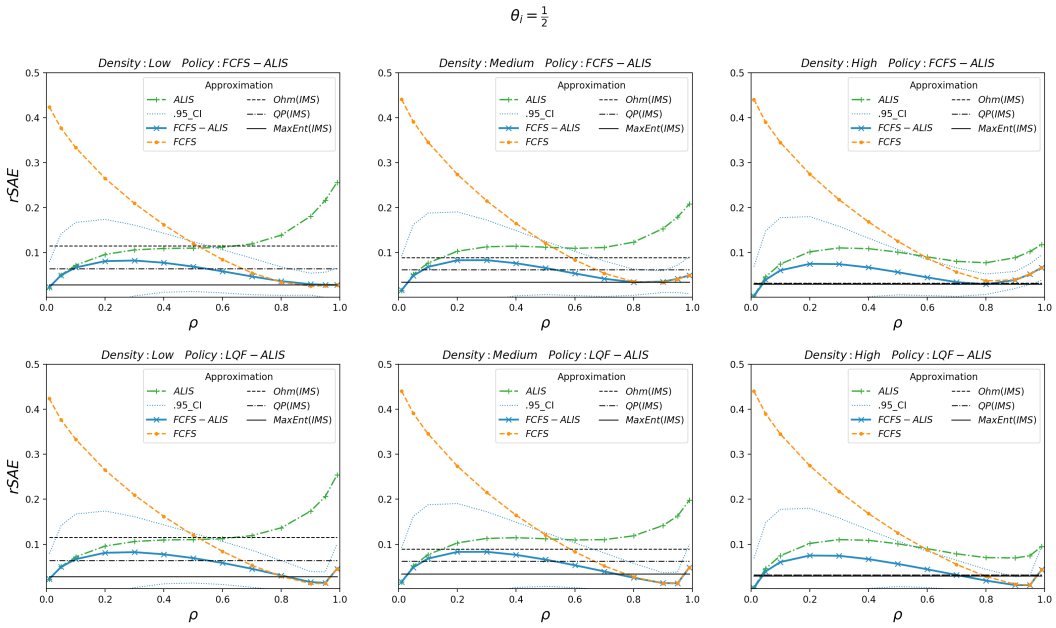


Figure 3.9: Error Rate by Traffic intensity for $\theta_i = .5, i \in \mathcal{I}$

to be a valid approximation for large-scale Erdős-Rényi and Torus graphs where CRP holds with an accuracy of 6% for both FCFS-ALIS and LQF-ALIS at high traffic intensities ($> .75$). The error rate is similar to the error rates of the quadratic approximation of [4] for infinite matching sequences and considerably better than the error rates of the Ohm's law approximation of [20]. For the Map graphs where the CRP does not hold valid approximations could not be obtained for utilizations below .99%, these results suggest that, when the CRP condition does not hold, the min-max-fair decomposition must be considered in deriving approximations of the matching rate. In summary, the approximation schemes developed in Chapter 3 provide the first scalable and robust method for approximating the matching rates of skill based parallel service systems with homogeneous service under both the FCFS-ALIS and LQF ALIS policies when the CRP holds. In the following Chapter we will demonstrate that these approximations also provide an understanding of the dynamics of the systems and enable the derivation of improved non-idling, non-preemptive policies policies.

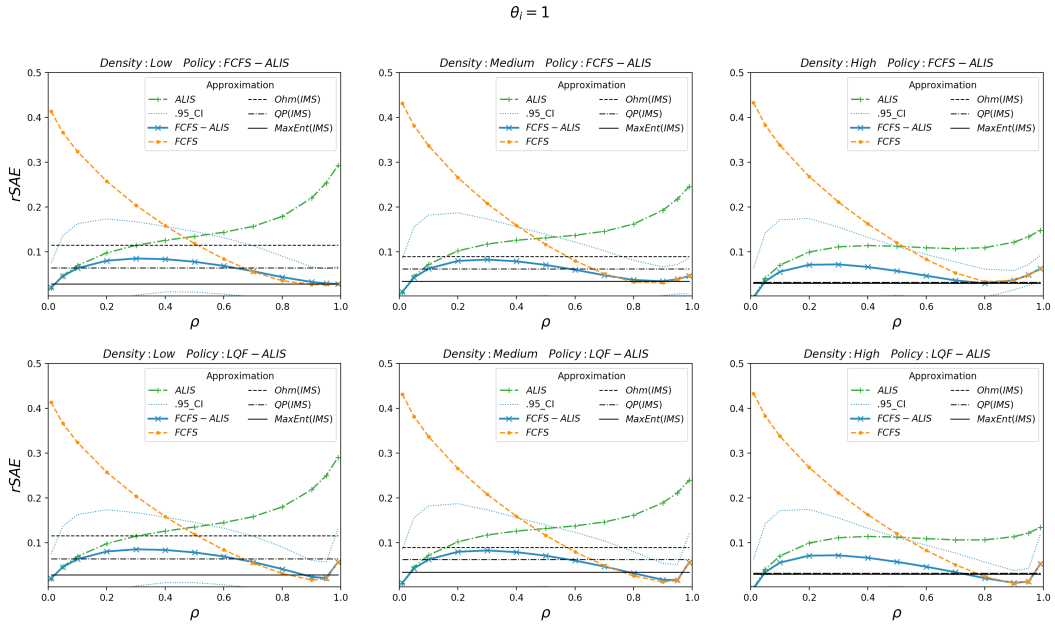


Figure 3.10: Error Rate by Traffic intensity for $\theta_i = 1, i \in \mathcal{I}$

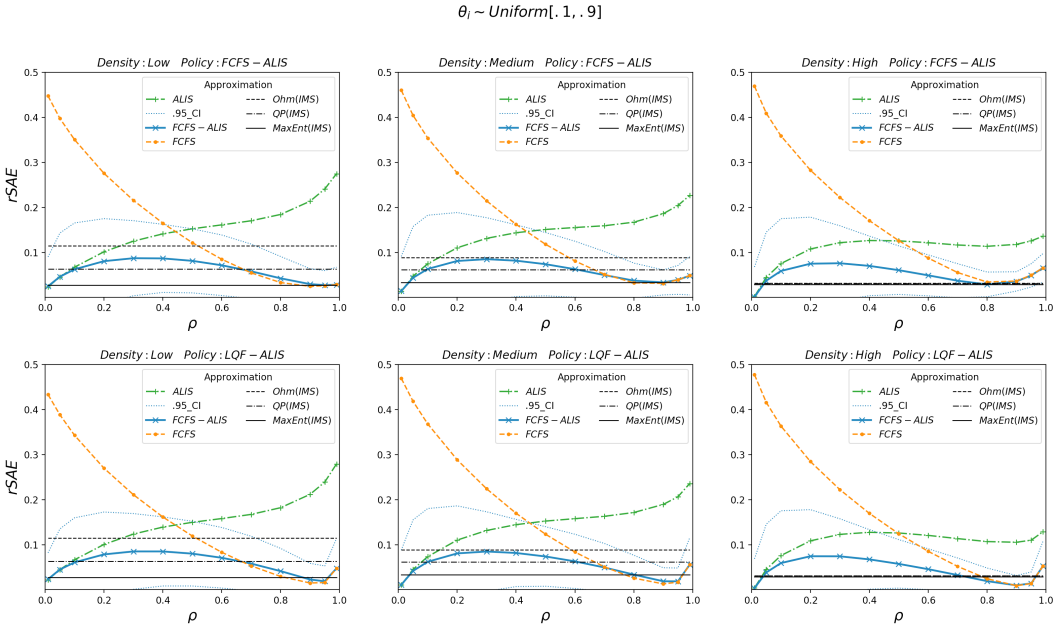


Figure 3.11: Error Rate by Traffic intensity for $\theta_i \sim Uniform[0, 1], i \in \mathcal{I}$

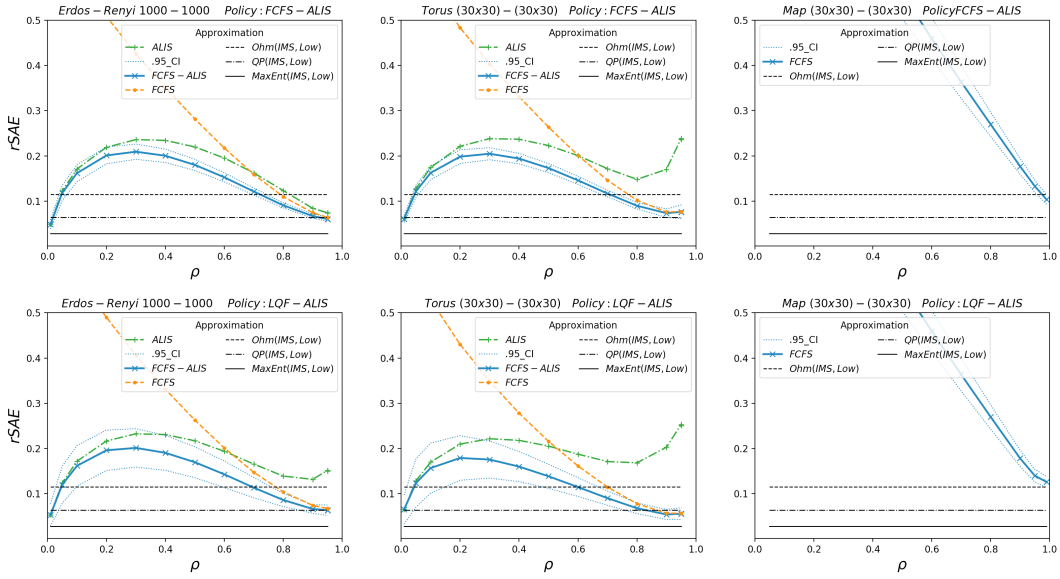


Figure 3.12: Error Rate by Traffic intensity for large scale SBPSSs

Chapter 4

Control Policies for Skill Based Parallel Service Systems

4.1 Motivation

The servers of an $M/M/c$ queue are assumed to be indistinguishable from one another and therefore a rule determining which server is chosen from among multiple available servers need not be specified. However, in the case of an SBPSS the servers are no longer assumed to be identical and hence one must specify the policy by which, when a single customer can be assigned to one of multiple servers, a server is chosen. One such common rule is the assign-longest-idle-server(ALIS) which states that the server that has been idle for the longest period of time will be chosen. As with any queueing system it is also necessary to specify how a server should choose a customer when multiple customers are waiting for service. In conjunction with the ALIS rule, the first-come-first-served (FIFO-ALIS) policy and longest-queue-first (LQF-ALIS) policy are the most commonly used in practice, and for good reasons. First, both are guaranteed to maintain finite queue lengths, i.e., to keep the system stable [2],[40], when doing so is feasible. Second, these methods can be easily implemented in practice as they do not assume any knowledge of the system parameters such as arrival and service rates and only require finding the maximum waiting time or queue length among customer classes or the maximum idle time amongst servers. Finally, each policy provides a certain sense of fairness from the perspective of the individual customer. In a system operating under a FIFO-ALIS policy no customer can cut in front of another when both can be served by the same server while under a LQF-ALIS policy the customer at the back of the longest queue can see that, even though it has the largest number of customers ahead of it, the situation is attended to by all servers that are capable of doing so. From the server perspective, if one equates idleness with rest, the ALIS policy assures that a resting server will not be assigned work when there is another server that has been resting for a longer time period and is capable of serving the customer. The question is therefore: Why, considering these favorable traits, would we wish to replace these methods

?. The answer is that despite providing a short term sense of fairness such policies may, in the long run, lead to an unfair distribution of waiting times across the customer classes and an unfair distribution of workload between the servers. The reason for this is that in a system operating under one of the aforementioned policies an individual customer, when afforded the option, will make use of any available server regardless of the impact that use may have on the other customer classes and similarly, a server will be assigned any job whenever it is the longest idle server amongst qualified servers regardless of the expected remaining idle time of other idle servers. In other words, these policies may be *fair* but they are not *courteous*. This can be demonstrated by example of the *increasing-N* system descried in Section 3.3 where illustration of such a system for $n = 5$ is given in Figure 3.6. Figure 4.1 shows the results of an experiment in which the *increasing-N* system is simulated under both the FIFO-ALIS and LQF-ALIS policies for size $n = 5, 10, 50, 100$ and $\rho = .85$; the resulting waiting times by class is plotted aginst the system size. and the utilization rates

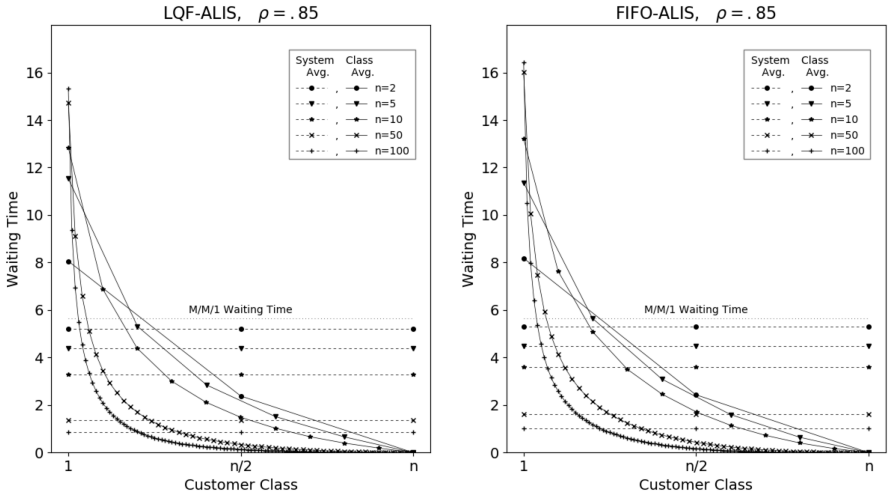


Figure 4.1: Waiting time by class in Increasing N system for $n = 2, 5, 10, 50, 100$

of the servers is similarly plotted in Figure 4.2 From the myopic viewpoint of any individual customer the system is *fair* in the sense that under the FIFO-ALIS policy, no customer *cuts* in front of another customer. That is, if customer A arrived before customer B but customer B was served before customer A it is only because the server that took customer B was not qualified to serve customer A. Similarly, under the LQF-ALIS policy, if the queue of class i_A customers was longer than that of the the i_B class customers but a customer of the i_B class was assigned next, it is only because that server that became available was not qualified to serve class i_A. The system is also *fair* from the myopic viewpoint of the servers, in the sense that anytime two servers may be chosen from, the server that has been idle for the longest time period is chosen. Despite satisfying this notion of fairness, as indicated clearly by the graphs in Figure 4.1, there is a striking imbalance between the waiting times experienced by

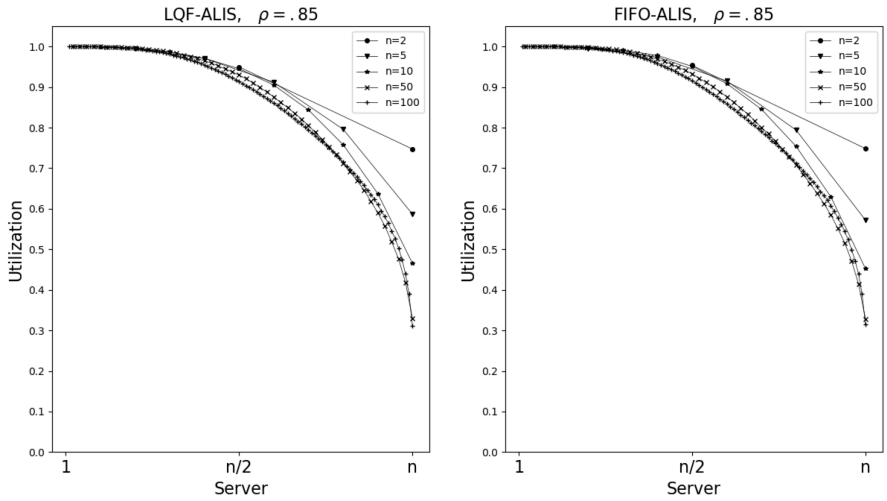


Figure 4.2: Utilization by server in Increasing N system for $n = 2, 5, 10, 50, 100$

different customer classes. The waiting time of the $i = 1$ customer class increases as n grows despite the fact that the average cycle time across all customer classes decreases with n. The imbalance in waiting time is the direct result of the imbalance in server utilization shown in Figure 4.2. This imbalance in utilization is caused by the fact the customers of classes $i > 1$ will utilize server $1'$ whenever the opportunity is afforded to them due to the random nature of the system. A customer of class 1 will only be served when it is either the longest waiting customer in the entire system, under the FIFO-ALIS policy, or when it is at the head of the longest queue in the LQF-ALIS policy. The n customer class on the other hand has a dedicated server. Similarly, server $1'$ can only become idle when there are no waiting customers in the system while server n' becomes idle upon completion of service, whenever the queue of class n customers is empty. An action that may be taken to address the waiting time imbalance in this SBPSS would be to restrict the service of a customer of class i by a server j' whenever $i > j$, and thus force each server to a utilization equal to the system traffic intensity ρ. If such a restriction is made the system decomposes into n identical $M/M/1$ queues and the waiting time of each customer class is given by $\rho \cdot (1 - \rho)^{-1}$, which for the case of $\rho = .85$ given in Figure 4.1 will result in a uniform waiting time of 5.67 time units, higher than the average waiting time for any $n > 1$ and yet the lowest average waiting time possible for customers of class 1. Hence, restricting the assignments produces a perfectly *fair* system from a long term perspective, but an unfair system from the short term perspective and, more importantly, increases the waiting time for most customer classes and the average waiting time of the system. Note that in the case where $\rho \to 1$ the only way to maintain system stability is to have the assignment rate between any customer class $i > 1$ and server $j < i$ tend to 0 and effectively balance the server utilization. However, as demonstrated by this example, when the overall system utilization is bounded away from 1, load balancing,

which is often considered as an objective in and of itself, can lead to poor performance. As depicted in Figure4.1, operating under either a FIFO-ALIS or LQF-ALIS policy results in lower average waiting times for most customer classes compared to the restricted system but also results in an increasingly *unfair* division of the average waiting time across customer classes. An improved control policy is thus one that would both reduce the average system waiting time compared to FIFO-ALIS or LQF-ALIS and result in improved *fairness* in the division of the waiting time across customer classes. The notion of improved fairness among customer classes first requires that we define a metric of fairness, one will be provided in Section 4.2. Having established a metric for fairness in Section 4.3 we use the Entropy Based approximation of Chapter 3 to gain insights and reveal a fundamental flaw inherent in the basic FIFO-ALIS and LQF-ALIS policies. Equipped with the new insight in Section 4.4 we further leverage upon our approximations to develop weighted versions of the standard policies and show their effectiveness in both balancing and reducing avg. waiting times using simulation experiments in Section 4.5. Finally, in Section 4.6 we suggest an extension of the weighted policies to systems with matching rewards and explain how the policies can be used to efficiently trade-off between long term avg. reward maximization and avg. waiting time and then go on to demonstrate this with simulation experiments in Section 4.7 .

4.2 The Waiting Time Gini Coefficient

The Gini coefficient [15], named after Italian statistician Corrado Gini (1884-1965), is a common measure of the inequality of wealth distribution within a population. The coefficient is usually defined using the Lorenz curve, named after American economist Max Otto Lorenz (1876-1959). The Lorenz curve plots the total wealth cumulatively owned by the bottom x percentile of a population. The passing of the Lorenz curve through a point (x, y) indicates that the bottom $x\%$ of the population, in terms of personal wealth, cumulatively posses y units wealth. In case of a completely equal distribution of wealth across the population the Lorenz curve is the straight line connecting between the points $(0, 0)$ and $(1, Y)$, referred to as the *line of equality*, where Y is the total communal wealth of the population. The Gini coefficient is defined as the ratio of the area that lies between the line of equality and the Lorenz curve (area I in Figure 4.3) over the total area under the line of equality (sum of areas I and II in Figure 4.3) so that a completely equal distribution of wealth will result in a Gini coefficient of 0 while a distribution in which a single member of the population posses the entire wealth will result in a Gini coefficient of 1. In order to use the Gini coefficient as a measure of the fairness of the waiting time distribution across classes we plot the Lorenz curve of cumulative waiting time of the population. For a given simulation experiment, let N_i be the number of class i customers served and let $N = \sum_{i\in\mathcal{I}} N_i$. We define $\hat{\lambda}_i = {}^{N_i}/_{N}$ as the proportion of customers served that are of class i; if the system is stable we can expect that $\lim_{N\to\infty} \hat{\lambda}_i = \lambda_i$. Let wt_k denote the waiting time of the k-th customer served and let WT_i be the average waiting time of customer class $i \in \mathcal{I}$ in a simulation experiment. The indices $k_1, \ldots, k_N$ are an ordering of customers by waiting time

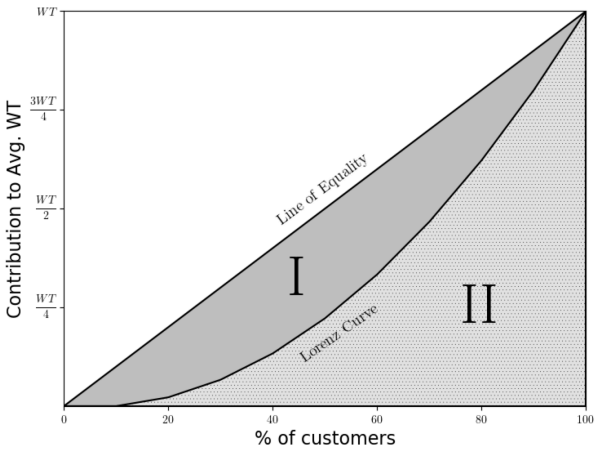


Figure 4.3: Lorenz Curve of Avg. WT contributions

such that $wt(k_1) \leq wt(k_2) \cdots \leq wt(k_N)$ and similarly the indices $i_1, \ldots, i_m$ are an ordering of customer classes by waiting time such that $WT(i_1) \leq WT(i_2) \cdots \leq WT(i_m)$. We define the cumulative waiting times on the customer and class levels by

$$\bar{\lambda}_{i_q} = \sum_{\ell=1}^{q} \lambda_{i_\ell} \tag{4.1}$$

$$\overline{WT}(i_q) = \sum_{\ell=1}^{q} \lambda_{i_\ell} WT_{i_\ell} \tag{4.2}$$

$$\overline{wt}(k_q) = \frac{1}{N} \sum_{\ell=1}^{q} wt_{k_\ell}. \tag{4.3}$$

We define the class level Lorenz curve is as the curve that passes through the points

$$(0,0) \to (\bar{\lambda}_{i_1}, \overline{WT}_{i_1}) \to \cdots \to (\bar{\lambda}_{i_m}, \overline{WT}_{i_m}). \tag{4.4}$$

Note that by definition $\bar{\lambda}_{i_m} = 1$ and $\overline{WT}_{i_m} = \overline{wt}_{k_N}$ is the average waiting time of all customers in the system, denoted by Wq. On a side note we mention that while in the typical wealth-based Lorenz curve of economics the most *well-off* individuals or groups are the ones in the right hand side of the curve, in our average waiting time contribution-based Lorenz curve it is the most unfortunate customers or customer classes with the longest waiting times the occupy that part of the curve. Based on this Lorenz curves we define a class level Gini index and we will say one service policy is fairer then the other if it has both lower class level Gini coefficients. Now that we have defined what constitutes a fairer service policy we can attempt to define a better matching policy.

4.3 What is Wrong with FIFO and LQF policies?

The approximations of (3.71) and (3.72) have an additional important and intuitive interpretation. This interpretation is best explained by an example of what may constitute as a bad approximation of the matching rates. Let us consider the following N-system of figure 4.4, where $\mathcal{I} = \{1, 2\}$, $\mathcal{J} = \{1', 2'\}$, $E = \{(1,'1), (2,'1), (2,'2)\}$ with $\boldsymbol{\lambda} = (.4, .4)$, $\mathbf{s} = (1, 1)$ and $\boldsymbol{\mu} = (.6, .4)$. If we approximate the matching by distributing the arrivals of each customer

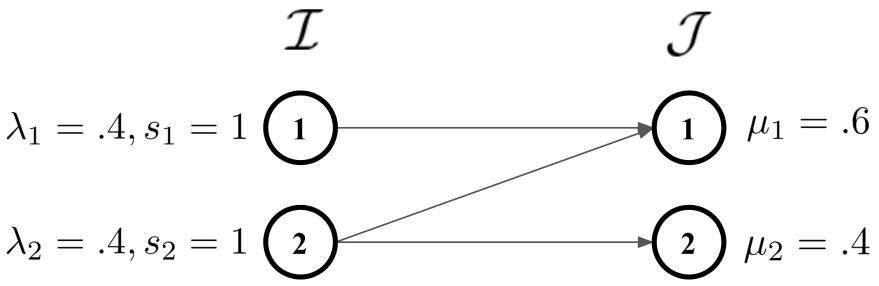


Figure 4.4: An N-system of size 2

class equally across the qualified servers we would obtain the following set of admissible matching rates:

$$\mathbf{r} = \begin{bmatrix} .4 & 0 \\ .2 & .2 \end{bmatrix}$$

Clearly, these matching rates, although admissible, can not be the matching rates of an SBPSS under any non-idling, non preemptive service policy. In order for such matching rates to be realized by the SBPSS it is required that customers of class 2 match with server 2' at the same rate as they match with server 1' despite the fact that server 1 is critically utilized and server 2 is idle 50% of the time. The reason why such matching rates are so clearly impossible is that any non-idling, non-preemptive service policy, merely by the fact that it is non-idling and non-preemptive, produces a natural load balancing mechanism that prevents the SBPSS from realizing such matching rates. It is reasonable to expect that under a non-idling, non-preemptive service policy, the matching rates of customers of type 2 to servers 1' and 2' will be approximately proportional to the portion of time each server spends idle. More generally, for an SBPSS operating under a non-idling, non-preemptive policy, we would expect that the rate at which the workload of a given customer class is removed by a qualified server to be approximately proportional to the portion of time the server is idle. This is exactly what is implied the approximation of (3.71) ,(3.72). For any $i \in \mathcal{I}$ and any $j, j' \in \partial(i)$ we have:

$$\frac{\eta_{ij}}{\eta_{0j}} = \frac{\eta_i \mu_j e^{V_i - W_j}}{\eta_0 \mu_j e^{V_0 - W_j}} = \frac{\eta_i e^{V_i}}{\eta_0 e^{V_0}} = \frac{\eta_i \mu_{j'} e^{V_i - W_{j'}}}{\eta_0 \mu_{j'} e^{V_0 - W_{j'}}} = \frac{\eta_{ij'}}{\eta_{0j'}}. \tag{4.5}$$

This relation in (4.5) is in fact equivalent to the Max Entropy approximation of (3.71) (3.72). It is easy to verify that, given an admissible set of workload removal rates for which (4.5) holds, the values

$$V_i = \log\left(\frac{\eta_{ij}}{\eta_{0j}\eta_i}\right), \quad i \in \mathcal{I} \tag{4.6}$$

$$W_j = -\log\left(\frac{\eta_{0j}}{\mu_j}\right), \quad j \in \mathcal{J} \tag{4.7}$$

satisfy the conditions in (3.71), (3.72). Although (4.6), (4.7) are an approximation and not an exact result, by shifting our perspective we can now draw insight from the approximation and use that insight to construct improved service policies. Instead of viewing the relation in (4.6), (4.7) from the perspective of an Input Queued (IQ) system in which the customers line up in queues by class (the order of input determines the order of the queue and hence the term) we may consider this from the perspective of an Output Queued(IO) system. In an OQ system with Markovian routing (recall that we have used this analogy previously in Chapter 2), a customer of class i is assigned immediately upon arrival to a server $j \in \partial(i)$ with probability r_{ij}/λ_i and joins a queue of customers waiting to be served by server j (the order of the queue determines the order of output and hence the term). In such an output queued system if we assume Poisson arrivals and exponential server dependent service times then each separate server along with its associated queue behave as an independent $M/M/1$. Therefore, the average length of the queue in front of the server j is given by

$$Lq_j = \frac{\sum_{i\in\partial(j)} \eta_{ij}}{r_{0j}} = \frac{\mu_j - r_{0j}}{r_{0j}}, \quad j \in \mathcal{J} \tag{4.8}$$

and, by Little's law, the average number of class i customers queued behind server $j \in \partial(i)$ is given by

$$Lq_{ij} = \frac{r_{ij}}{r_{0j}}, \quad (i,j) \in E \tag{4.9}$$

From the OQ system perspective the result of (4.6), (4.7) states that, if the maximum entropy assignment rates were used to route the customers, the average number of customers of class i in any queue $j \in \partial(i)$ is the same. Hence we can define, *for all* $j \in \partial(i)$,

$$Lq_{ij} = \frac{r_{ij}}{r_{0j}} = Lq_i, \quad i \in \mathcal{I}. \tag{4.10}$$

If we consider this property, still from the OQ system perspective, an inherent flaw becomes apparent: The customers of a given class $i \in \mathcal{I}$ are assigned to queues in a such a way that there is an equal average number of class i waiting customers in each qualified server queue. This property may cause a severe imbalance in the lengths of the server queues. If we consider the example of the N-increasing system, the avg. number of waiting class n

customers is identical in both the n' server queue, which is completely dedicated to serving customers of class n, and the server $1'$ queue which servers all customer classes in the system and is the single server qualified to server customers of class 1. The result of maximizing (3.68) subject to constraints (3.69), (3.70) is that the term Lq_{ij} is the same across all $j \in \partial(i)$ despite the fact that the queue lengths themselves $Lq_j, j \in \mathcal{J}$ may be imbalanced. Figure 4.5 demonstrates the Avg. state of an OQ *increasing-N* system with Markovian and FCFS-ALIS matching rate routing probabilities. The constraint (3.72) implies that Lq_j is strictly

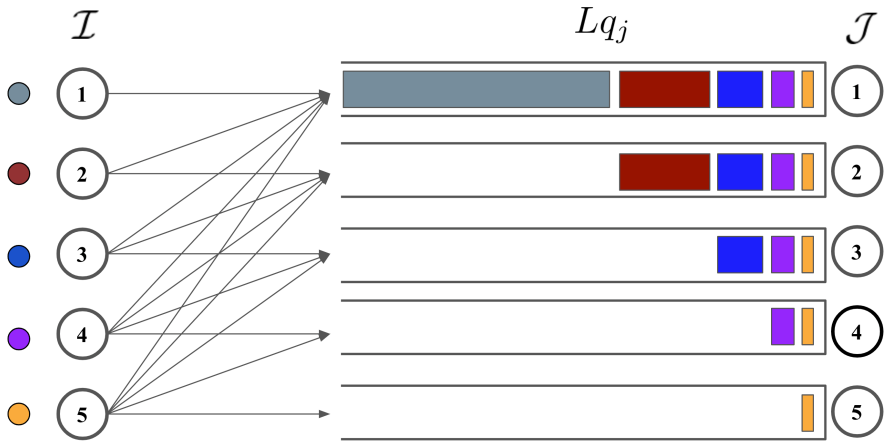


Figure 4.5: An OQ systems with FCFS-ALIS matching rates as routing probabilities

determined by r_{0j} and hence if we obtain a more uniform distribution of $\{r_{0j}, j \in \mathcal{J}\}$ across the capacity of the servers we can obtain a more uniform distribution of $\{Lq_j, j \in \mathcal{J}\}$. The most uniform distribution of idleness $\{r_{0j}, j \in \mathcal{J}\}$ across capacity (not across individual servers) is by definition the one obtained by a *min-max-fair* assignment. Let us isolate the RHS of (3.68) and define

$$\mathcal{H}_0(\mathbf{r}) = \sum_{j \in \mathcal{J}} r_{j0} \log\left(\frac{r_{j0}}{\mu_j}\right) \tag{4.11}$$

and make the following observation:

Lemma 4.3.1. *Anset of matching rates* $\mathbf{r}$ *is a min-max-fair assignment if and only if* $\mathbf{r}$ *is an optimal solution of the convex program:*

$$MaxEnt_0(G): \quad \mathcal{H}_0(\mathbf{r}) \tag{4.12}$$

$$\text{subject to} \quad \sum_{j \in \partial(i)} r_{ij} = \lambda_i \qquad ,\forall i \in \mathcal{I} \tag{4.13}$$

$$\sum_{i \in \partial(j)} r_{ij} + ri_{0j} = \mu_j \qquad ,\forall j \in \mathcal{J} \tag{4.14}$$

Proof. Functions of the form $g(x) = -\alpha x \log(x)$ are strictly concave when $\alpha > 0$ and $x \in (0, 1]$ since $g''(x) = -\frac{\alpha}{x} < 0$. Hence the objective function is the sum of strictly concave functions and therefore is itself strictly concave with respect to the variables $\{r_{0j}, j \in \mathcal{J}\}$. The feasible region is defined by a finite set of linear equality and inequality constraints and we can therefore conclude that (4.12) has a unique optimal solution in terms of the $\{r_{0j}, j \in \mathcal{J}\}$ variables and that each such solution must be a KKT point (see [10]). We will now use the structural properties of *min-max-fair* assignments from Lemma 2.1.1 to define KKT constants (x_i, y_j, z_{ij}) such that, when paired with any *min-max-fair* assignment $\mathbf{r}^f$,the KKT optimality conditions for (4.12) are met. Primal feasibility follows immediately form the fact that a *min-max-fair* assignment is by definition a feasible one. The stationarity condition for (4.12) is given by

$$-(1 + \log(r^f_{0j})) = -y_j, \quad \forall j \in \mathcal{J}, \tag{4.15}$$

$$0 = -y_j + x_i - z_{ij}, \quad \forall (i, j) \in E. \tag{4.16}$$

The two conditions can be reduced using basic operations to:

$$(1 + \log(r^f_{0j})) = x_i - z_{ij}), \quad \forall (i, j) \in E. \tag{4.17}$$

We now use the function ϕ defined in (2.13) to define the following values for $x_i, z_{i,j}$:

$$z_{i,j} = \left(\log(r^f_{0\phi(i)}) - \log(\pi_{0\phi(j)})\right), \quad \forall (i, j) \in E, \tag{4.18}$$

$$x_j = \left(1 + \log(r^f_{0\phi(i)})\right), \quad \forall i \in \mathcal{I}. \tag{4.19}$$

It is easy to verify that in case of a *min-max-fair* assignment $\mathbf{r}^f$, where $\rho_j(\mathbf{r}^f) = \rho_{\phi(i)}$ for all $j \in \mathcal{J}$, the stationarity condition holds for this choice of constants. The complementary slackness condition for (4.12) requires that

$$z_{ij} \cdot r^f_{ij} = 0, \quad \forall (i, j) \in E. \tag{4.20}$$

To see that this condition holds for any *min-max-fair* assignment note that by Lemma 2.1.1 for any *min-max-fair* assignment $r^f_{ij} > 0 \Rightarrow \phi(i) = \phi(j)$ which in turn implies that $z_{ij} = 0$ by (4.19) and hence the condition is met. Finally, the dual feasibility condition requires that

$$z_{ij} \geq 0, \quad \forall (i, j) \in E. \tag{4.21}$$

Observe that in order for z_{ij} defined by (4.19) to be strictly negative we must have $\rho_{\phi(j)} < \rho_{\phi(i)}$ which implies that $j \in J_k(\mathbf{r}^f)$ and $i \in I_\ell(\mathbf{r}^f)$ for some $k < \ell$ which is a contradiction as theorem 2.1.2 states that for any *min-max-fair* assignment there can be no qualifications between the customers of I_ℓ and servers in J_k. We conclude that $z_{ij} \geq 0$ for all $(i, j) \in E$. Hence, with constants x_i, z_{ij} defined by equations (4.18),(4.19) we have proven that any *min-max-fair* assignment is an optimal solution to the maximum server entropy problem in (4.12). The converse follows immediately from theorem (2.1.2) given the strict convexity of the objective in (4.12) with respect to the variables $r_{0j}, j \in \mathcal{J}$. □

□

An immediate corollary of Lemma 4.3.1 is that an optimal solution of (4.12) will produce, in an output queued system a min-max-fair distribution of queue lengths. In the case of homogeneous servers a min-max-fair distribution of queue lengths also implies a minimum average number of customers in the system, this is not the case for a system with server with heterogeneous rates. Despite the fact that a min-max-fair assignment minimizes the average number of customers in the system for a homogeneous server output queued system, obtaining such assignment rates in the input queued non-idling,non preemptive SBPSS may be neither optimal nor feasible. As an example let us consider the *increasing-N* system. A *min-max-fair* assignment for the *increasing-N* system is given by:

$$\mathbf{r}^f_{ij} = \begin{cases} \lambda_i, & \textit{If } i = j \\ 0, & \textit{Otherwise} \end{cases} \tag{4.22}$$

The *min-max-fair* assignment in eq.(4.22) indicates a preference for assigning customers of class i to server i and yet, for any $n > 2$ we have, $\mathbf{r}_{n1} = p_{21} = 0$ despite the fact that customers of class can be served by all servers while customers of class 2 may only be served by server 1 and 2. The dedication of server i to customer class i implied by the *min-max-fair* assignment is the only feasible assignment rate matrix for the system when $\eta = 1$ and hence any stable service policy such as LQF-ALIS [40] or FIFO-ALIS [2] will converge to $\mathbf{r}^f$ as $\eta \to 1$. However, for any $\eta << 1$ it is clear that these rates are not feasible for the *increasing-N* system under any non-idling policy. Regardless of which non-idling policy is used, if the system is stable, the long term average proportion of the time server 1 spends serving customers of class 1 must be η, leaving a $1 - \eta > 0$ fraction of the time in which server 1 is not serving class 1 customers. If server 1 is not serving customers of class 1 than either it is serving a customer of a different class $i > 1$ or it is idle and any otherwise unassigned customer of a class $i > 1$ must, by the non-idling nature of the service policy, be assigned to server 1. Furthermore, by the same principle, even if idling service policies are allowed, obtaining the *min-max-fair* assignment rates requires that (i, j) assignments be avoided whenever $i > j$ and the system must thus be divided into n separate single server systems. In such case, as shown in Figure 4.1 for the case $\eta = .85$, the average waiting time in the system will increase compared to a LQF-ALIS or FIFO-ALIS policy for any $n \geq 2$. The problem with the LQF-ALIS and FCFS-ALIS policies as implied by output-queued analogy and by the example of the N-system is that although these non-idling, non-preemptive policies are asymptotically load balancing, for sub-critical systems they can still create severe load imbalances across the servers which may result in a similar imbalances of the waiting times across the customer classes.

4.4 Weighted FCFS-ALIS and LQF-ALIS policies

On the one hand, the approximation provided by (4.6), (4.7) exposes the inherent flaw of non-preemptive, non-idling policies; on the other hand, as will show in the section, it can also provide a basis for constructing improved service policies that mitigate those flaws. Recall

that the approximate matching rates of a system under an FCFS-ALIS or LQF-ALIS policy may be obtained by solution of the convex program in (3.68),(3.69), (3.70). Let us now observe the two terms that comprise the objective (3.68) separately. Lemma 4.3.1 implies that maximizing the right hand term of (3.68):

$$\sum_{j\in\mathcal{J}} r_{0j}\log\left(\frac{r_{0j}}{\mu_j}\right) \tag{4.23}$$

subject to constraints (3.71), (3.72) will result in a *min-max-fair* workload assignment. If the CRP condition holds this will imply a uniform distribution of workload across all the servers. Hence we can view the right side of the objective as striving to balance the loading across the servers. If we now consider maximizing only left term of (3.68)

$$\sum_{i\in\mathcal{I}}\sum_{j\in\partial(i)} r_{ij}\log\left(\frac{r_{ij}}{\mu_j}\right) \tag{4.24}$$

it is easy to see that if the solution

$$r_{ij} = \frac{\lambda_i}{|\partial(i)|}, \quad \forall i\in\mathcal{I}, j\in\partial(i) \tag{4.25}$$

is feasible it will be the optimal solution as it is the global optimum of the unconstrained function. Furthermore, if $\mathbf{r}^*$ is an optimal solution when maximizing only the left side term then for any customer class $i\in\mathcal{I}$ and any $j,j'\in\partial(i)$ we have that $r^*_{ij'} > r^*_{ij} \rightarrow r_{0j} = 0$, otherwise, one can increase r^*_{ij} by some $\epsilon_j > 0$ and decrease $r^*_{ij'} > \epsilon_{j'} = \epsilon\left({}^{\mu_j}/_{\mu_{j'}}\right) > 0$ and thus increase the objective value. The left side term of (3.68) can thus be viewed as striving the balance the workload of each customer class evenly across all of its available service capacity. Recall that in a fully qualified system with $E = \mathcal{I}\times\mathcal{J}$ the workload of each customer class, by symmetry, will be proportionally divided across all servers according to their capacity, therefore, the left hand side can also be thought of as striving to maximize the effective number of servers each customer class uses. The matching rates of a high utilization SBPSS under an FCFS/LQF-ALIS policy can thus bee thought of as the result of the balance of these two terms. As we have shown, this may often result in an imbalanced server workload distribution. A key observation is that by assigning an increased weight to the right side term of (3.68) while assigning a lower weight to the left side term one can expect to obtain a set of admissible matching rates that induces an improved balance of server workloads. To choose an appropriate set of weights we first note that the impact of server workload balance on customer class waiting time balance depends upon the loading of the system. In a light traffic or low traffic intensity system an imbalance in server workload is not likely to cause an imbalance in waiting times as most arriving customers are still likely to find idle servers while for a heavy traffic of high traffic intensity system even a small imbalance of server workloads can cause a large imbalance of the customer class waiting times. Hence, a natural choice will be to weigh each term by the overall system utilization. However, as we have

shown in Chapter 2, this utilization value is not necessarily representative of the utilizations of every server. In case the CRP condition does not hold, some subsets of servers must have a higher workload under any feasible assignment. Therefore, we argue that each separate term in the summation on both the right and left side term of (3.68) should be weighted according to the *min-max-fair* utilization associated with the server. Consider the convex optimization

$$MaxEnt(G,\bar{\rho}): \quad \max_{r\in r_E, r_0\in\mathbb{R}_n^+} \mathcal{H}_{\bar{\rho}}(r)+\mathcal{H}_{\bar{\rho},0}(r_0) \tag{4.26}$$

$$\text{subject to} \quad \sum_{j\in\partial(i)} r_{ij}=\lambda_i \qquad ,\forall i\in\mathcal{I} \tag{4.27}$$

$$\sum_{i\in\partial(j)} r_{ij}+r_{0j}=\mu_j \qquad ,\forall j\in\mathcal{J} \tag{4.28}$$

where

$$\mathcal{H}_{\bar{\boldsymbol{\rho}}}(\mathbf{r})=-\sum_{(i,j)\in E}(1-\bar{\rho}_j)r_{ij}\log(r_{ij}) \quad \text{and} \quad \mathcal{H}_{\bar{\boldsymbol{\rho}},0}(\mathbf{r})=-\sum_{j\in\mathcal{J}}\bar{\rho}_j r_{0j}\log(r_{0j}) \tag{4.29}$$

and $\bar{\rho}_j$ is the *min-max-fair* utilization of server $j\in\mathcal{J}$ as defined and in Theorem 2.1.2. Let $\hat{\mathbf{r}}^*$ be the optimal solution of (4.26),(4.27),(4.28) and let $\mathbf{r}^*$ be the optimal solution of (3.68),(3.70),(3.69). Both are admissible matching rates yet the first, if used as routing probabilities in an output queued system with Markovian routing, would induce an improved server workload balance compared to the second. The question thus becomes: while the matching rates $\mathbf{r}^*$ can be approximately obtained by using a FCFS/LQF-ALIS policy how can the information provided by $\hat{\mathbf{r}}^*$ be used to create improved non-idling, non-preemptive policies. The answer we provide is that the distribution of the avg. number of waiting customers in an OQ system, the same one that we used to demonstrate the flaws of the FCFS/LQF-ALIS policies can also be used to create improved weighted FCFS/LQF-ALIS policies. Let Lq_j, Lq_{ij} and $\hat{Lq}_j, \hat{Lq}_{ij}$ be the avg. queue length and average number of class i customers in the queue for queue j of an OQ system as defined in (4.8),(4.9) the first defined by the routing probabilities $\mathbf{r}^*$ and the second defined using routing probabilities $\hat{\mathbf{r}}^*$. In order to derive a weight for an assignment $(i,j)\in E$ we consider the relative change in the average portion of class i customers in the server j queue in the OQ system when going from $\mathbf{r}^*$ to $\mathbf{r}^*$. This ratio is given by:

$$w_{ij}=\frac{\hat{Lq}_{ij}/\hat{Lq}_j}{Lq_{ij}/Lq_j}, \quad (i,j)\in E \tag{4.30}$$

These weights can now be applied to create weighted FCFS, LQF and ALIS policies as follows:

- **wFCFS:** *Server j will take longest waiting customer from queue i' where*

$$i'=\text{argmax}\{w_{ij}W_i(t)|i\in\mathcal{I}\}$$

- **wLQF:** *Server j will take longest waiting customer from queue i' where*

$$i' = \text{argmax}\{w_{ij}Q_i(t)|i \in \mathcal{I}$$

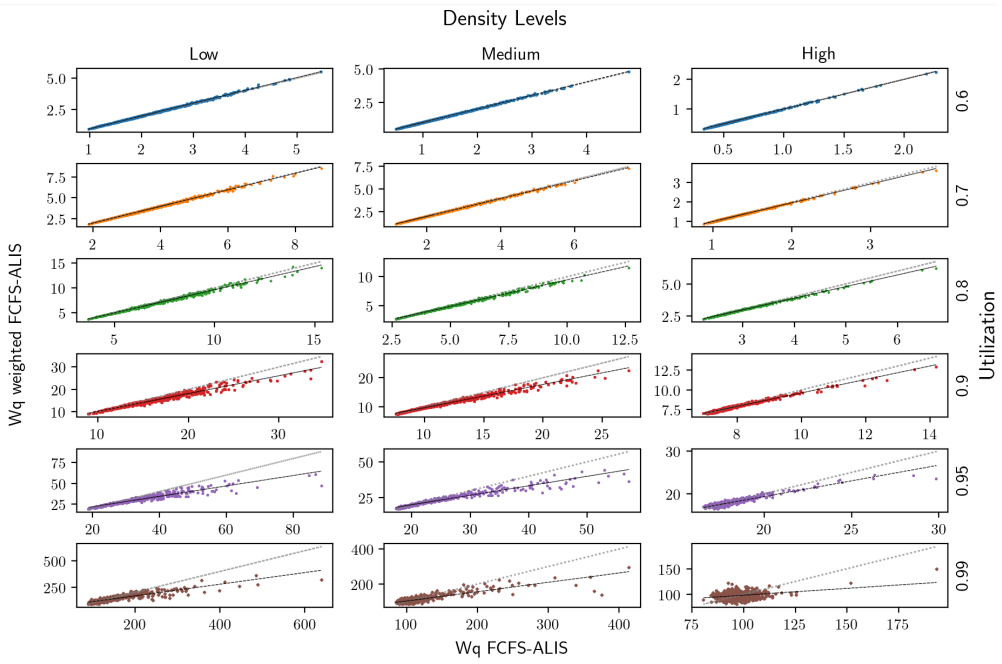
- **wALIS:** *Customer j will choose queue j' where*

$$j' = \text{argmax}\{w_{ij}I_j(t)|j \in \mathcal{J}\}$$

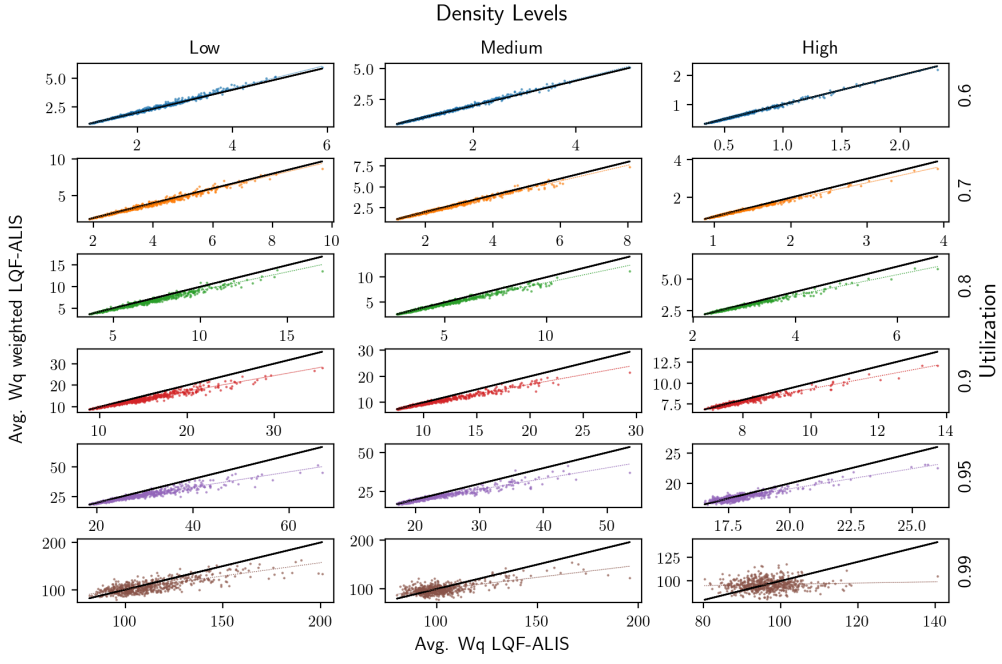
The basic idea here is that if balancing the workload of the servers implies increasing the relative portion of class i customers in queue j while decreasing the relative portion of class i' customers then, when given two possible assignments $(i,j),(i',j)$ class i should be favoured by server j. Similarly, if balancing the workload of the servers implies increasing the relative portion of class i customers at the server j queue while decreasing the relative portion of class i customers at the server j' queue, then given two possible assignments $(i,j'),(i,j')$ a class i customer should favour server j. As we will show in the following section with simulation experiments these weighted policies can provide a significant improvement over the standard FCFS/LQF-ALIS policies.

4.5 Simulation Experiments - Weighted Policies

In this section we use the same randomly generated graphs that were used to estimate the accuracy of the approximations in Chapter 3. This is only fitting as the weighted matching schemes in this chapter are based on the approximations of Chapter 3. For a proper comparison of the weighted polices to the standard ones we observe the systems only under medium to high traffic intensities. A system with low traffic intensity will not experience congestion and any improvement of the non-idling matching policy will have little to no impact on the system performance. In the examples of Map graphs the CRP condition does not hold and hence the mean traffic intensity across the system can not be achieved by all servers. Hence the utilization value specifies not the systems utilization but rather the maximal element of the systems *min-max-fair* utilization sequence. In all cases we are interested in two key performance measures, the avg. waiting time in the system and the distribution of that waiting time across customer classes as measured by Gini score. Figures 4.6a and 4.6b compare the waiting times under the standard FCFS/LQF policy against those under the weighted policies as observed in simulations of the randomly generated systems from the experiments of Section 3.1. Each small simulation consists of 10^5 customer arrivals and is repeated 30 times. The standard deviation of. the avg. waiting is less than .5% of the mean for all experiments. The weighted polices do not outperform the standard policies on all cases, however they do consistently outperform the standard polices for SBPSS with long waiting time, especially for low density SBPSS. The improvement due of the use of the weighted policy increases as the waiting time of the system under the standard policy increases. The two dark line in Figures 4.6a and 4.6b describe the line of equality and the regression lines fit $Wq_{FCFS-ALIS}, Wq_{LQF-ALIS}$ to $Wq_{wFCFS-wALIS}, Wq_{wFCFS-wLQF}$

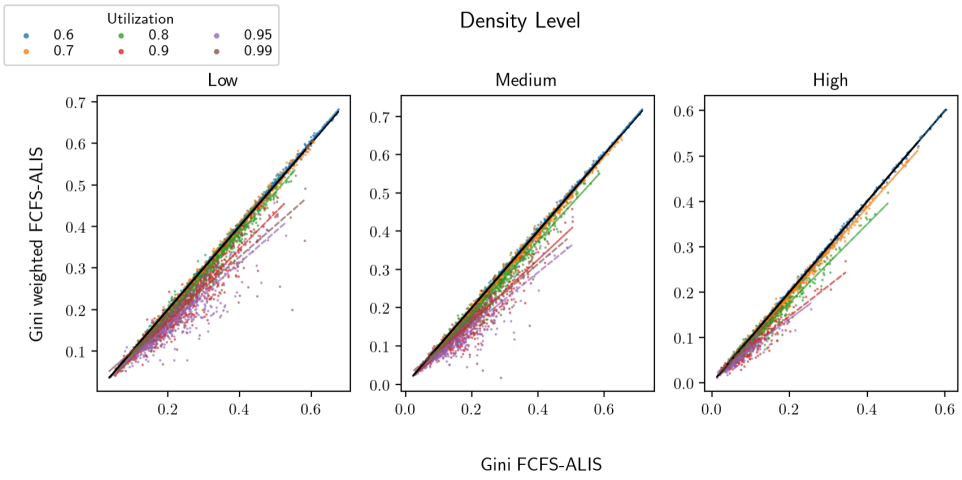


(a)

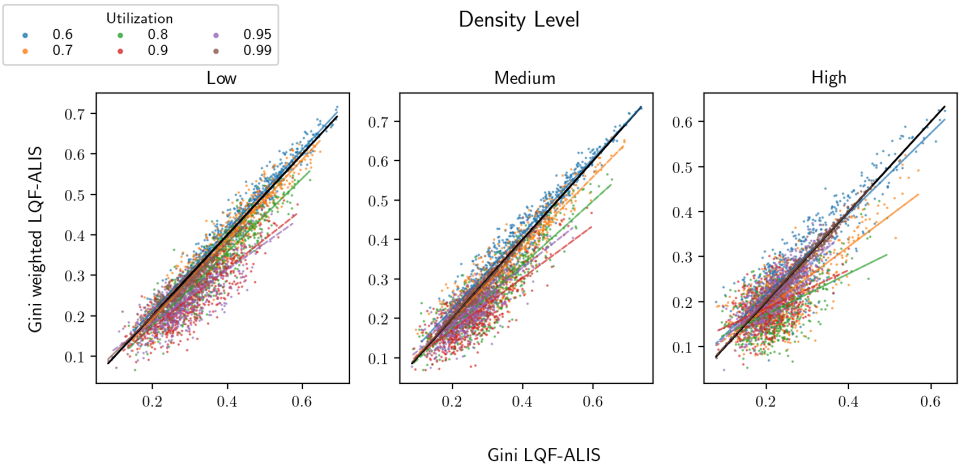


(b)

Figure 4.6: Avg. Wq for small graphs under weighted and standard police



(a)



(b)

Figure 4.7: Gini score in small graphs under weighted and standard police

respectively, one can observe that, for utilization levels of .8 and above, the regression line is discernibly below the line of equality. Similar results can be observed when comparing the Gini score obtained under both policies in Figures 4.7a and 4.7b. The weighted polices seems to have a greater impact on those SPBSS that have higher Gini scores and this difference is most prominent for the low density graphs. The similarity in the impacts of the weighted polices on both performance measures is not coincidental as plotting the average waiting times against the Gini Score under both weighted and Unweighted policies in Figure 4.8 shows that the two are correlated. Hence for the small sized SBPSS we can conclude that the weighted polices are more robust than the standard ones and reduce both the waiting time and the variance of waiting time amongst customer classes especially under high traffic

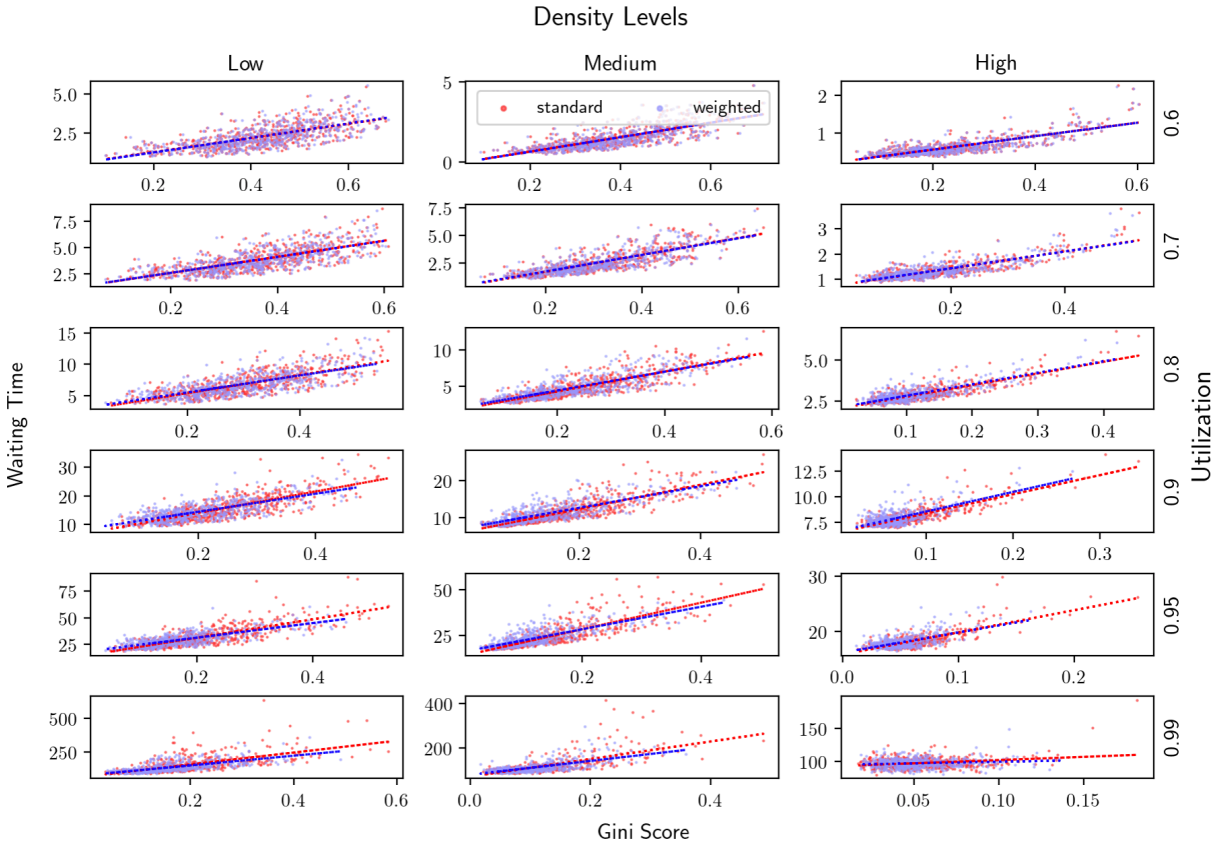


Figure 4.8: Avg. Wq vs Gini score for small graphs under weighted and standard polices

intensities ($\rho \geq .8$). Next, we turn to look at the larger scale graphs of section 1.5. For these experiments the same 30 instances of Chapter 3 we used for every structure. Each data point represents 30 repetitions of a simulation experiment with 10^7 customer arrivals. The Erdos-Renyi graphs are, as discussed in Section 1.5, expander graphs and hence as the scale of the graph increases the CRP condition is likely to hold and workload can be efficiently transferred between the servers when the system is congested. As a result, it can be seen in Figure 4.9 that although the wFCFS-wALIS policy does improve the performance of the Erdos-Renyi systems it is not a considerable improvement. The reason becomes more apparent if we observe the Gini score comparison in Figure 4.10 where we can see that for the Erdos-Renyi graphs as utilization and subsequently waiting times increase the distribution of the waiting times across customer classes tends to uniform with a low Gini score and the wFCFS-wALIS policy that is derived for the purpose of balancing the waiting times across customer classes has little to contribute. However, this is not the case for tours graphs which, although generated so that the CRP condition will hold, are not expander graphs. Figure 4.11 shows that the performance of the wFCFS-wALIS policy offers a considerable

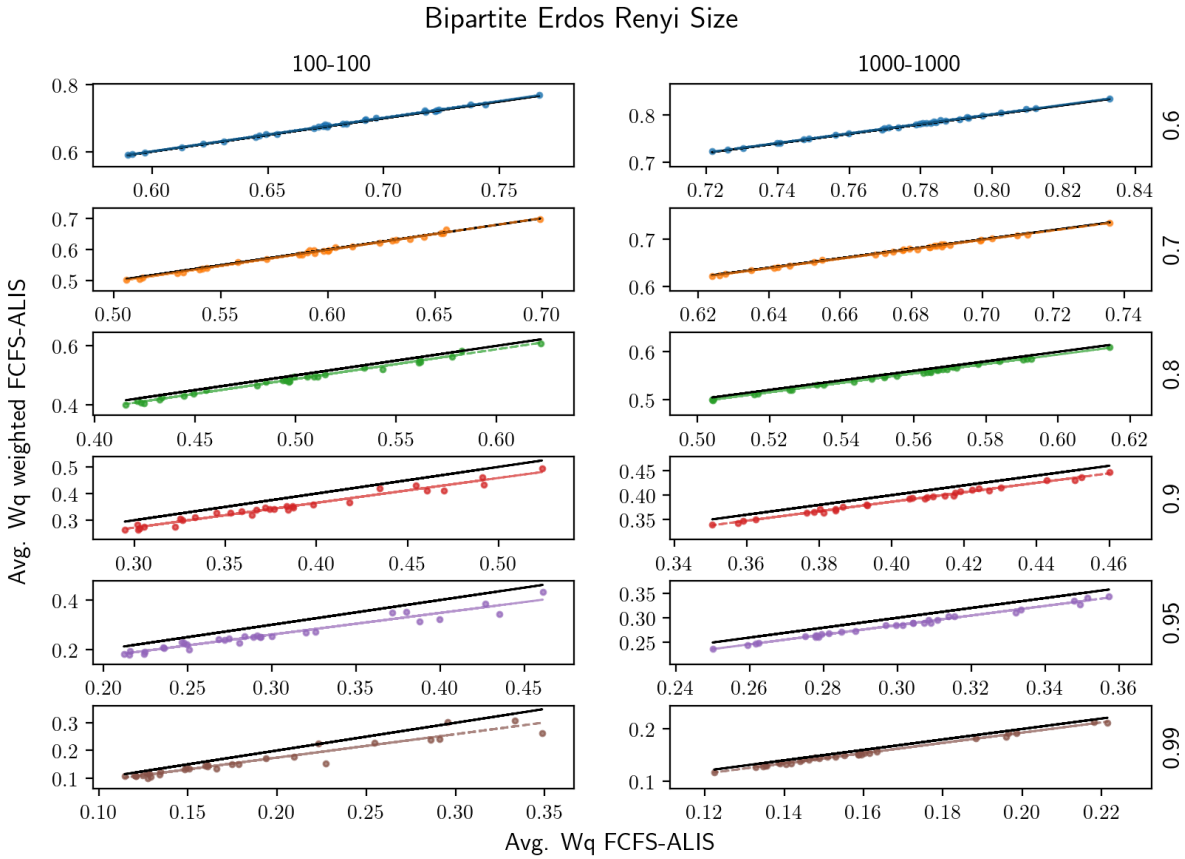


Figure 4.9: Avg. Wq in Erdős-Rényi graphs under weighted and standard police

improvement over the standard FCFS-ALIS policy. The cause of this improvement can be clearly seen in Figure 4.12 where we can see that for any utilization level of .8 and above the wFCFS-wALIS policy dramatically reduces the variance in the waiting times across customer classes as expressed by the Gini score. Note that this reduction in customer class waiting time variance occurs despite the fact that the CRP condition holds and a uniform assignment of the workload to the servers is feasible in an output queued system. Similar results can be observed for the of map graphs, here the CRP condition does not hold and a *min-max-fair* assignment does not balance the utilizations across the servers, as such the utilization values specified in the Figures 4.13 and 4.14 pertain to the minimal maximum utilization of the graph, i.e ρ_1 of the *min-max-fair* utilization sequence of the graph as defined in Theorem 2.1.1. In Figure 4.13 we can observe that for map graphs with $\rho_1 = 0.6$, which implies that all other customer/class other than I_1, J_1 have a utilization lower than .6, the use of a wFCFS/wLQF-wALIS may degrade the performance of the system, this can be expected as for system with little congestion there is no need to promote customers ahead of others that have longer waiting times or are in a larger queue as this will not

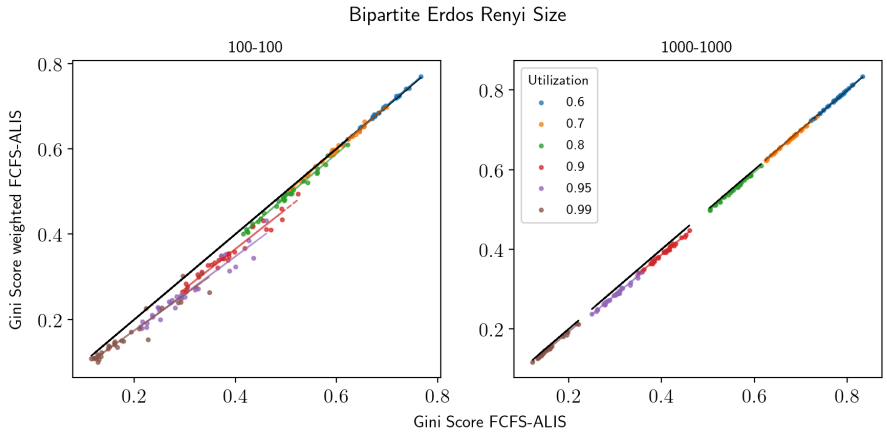


Figure 4.10: Gini score in Erdős-Rényi graphs under weighted and standard polices

likely reduce subsequent congestion. Nonetheless, as ρ_1 increase it becomes advantageous to prevent unnecessary congestion at the more utilized servers and the wFCFS/wLQF-wALIS policies can be seen to considerably improve the performance of the system especially for system with high waiting times. An interesting aspect is that while the reduction in waiting time offered by the weighted policies is clear, the Gini Score is less consistent, especially when comparing the LQF policies. However if we plot the ratio of the waiting times under weighted and standard polices against the ratios of the Gini scores in Figure 4.15 we can see that in all cases where the Gini score is lower under the standard policy the waiting time is considerably higher. This is a known disadvantage of the Gini score which may degrade even if the waiting times of all customer classes are improved. To conclude, the experiments clearly indicate the potential of applying the weighted ALIS/LQF polices over the standard ones for systems under high or heavy traffic intensities with a striking improvement achieved in SBPSS where the underlying compatibility graph is not an expander graph.

4.6 Weighted Policies for SBPSSs with Matching Rewards

In the previous section we demonstrated the use of weighted FCFS/LQF polices to reduce delay and customer class waiting time variance. In this section we wish to extend the use of weighted polices to cases where a matching of a customer and server incurs a reward/cost. In such cases a controller may wish to trade-off between maximizing the long term average reward and minimizing the waiting times of customers in the system. Let us consider a set of rewards $c_{ij}, (i,j) \in E$ associated with every compatible pair of customer class and server. In [44] the authors prove that, given that the SBPSS is feasible, this objective can be obtained with a "Greedy Primal Dual" algorithm that at each assignment decision greedily maximizes

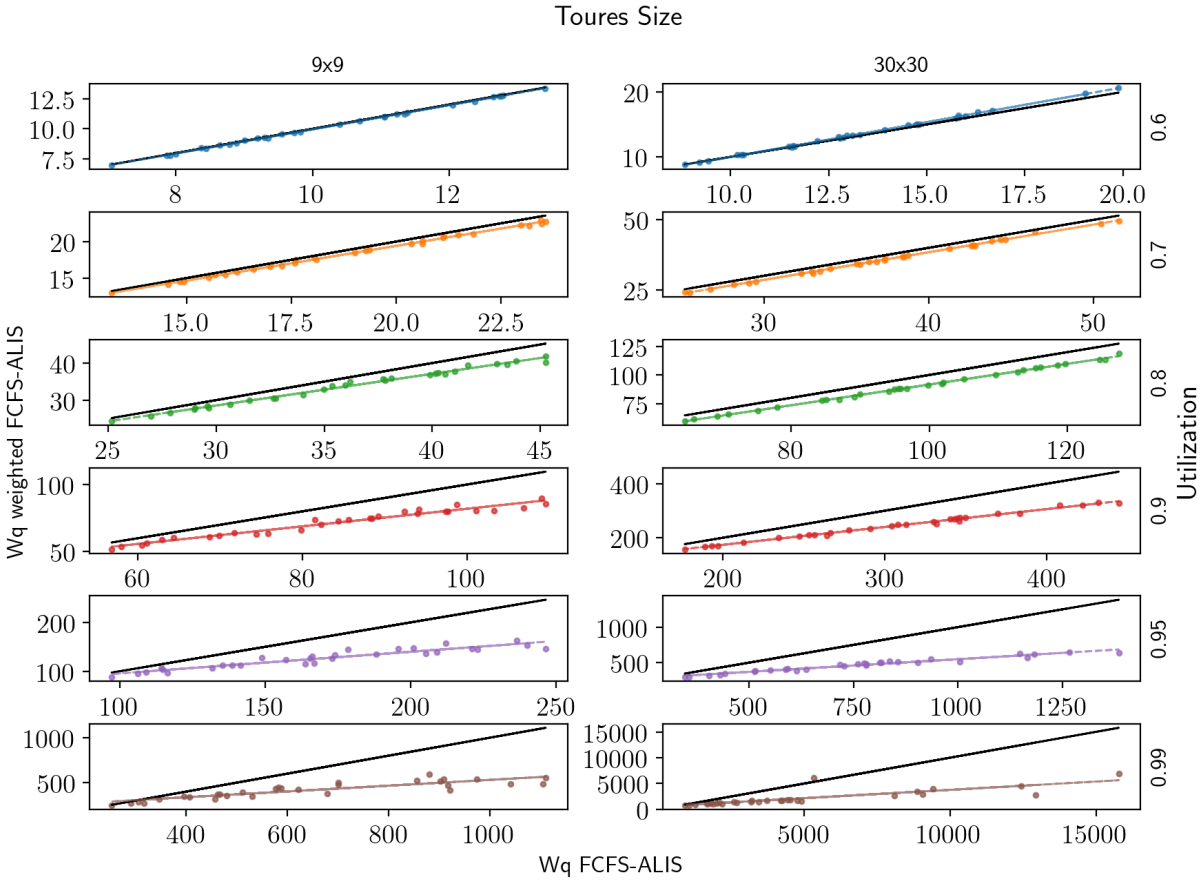


Figure 4.11: Avg. Wq in Torus graphs under weighted and standard polices

a sum of the assignment utility function (Primal) and a penalty on the queue lengths (Dual). However, maintaining the system stable only implies that waiting times are finite and does not guarantee any level of service. Furthermore, the method does not provide a means to efficiently trade-off long term average reward rate and the customer class waiting time and variance. Let us now consider the problem of maximizing the long term average reward in the output queued system with Markovian routing. This will result in the following LP:

$$\max_{r_{\in} r_E, \in \mathbb{R}_n^+} \mathcal{Z}_c(\mathbf{r}) = \sum_{(i,j)\in E} c_{ij} r_{ij} \tag{4.31}$$

$$\text{subject to} \quad (3.71),\ (3.72) \tag{4.32}$$

$$nonumber \tag{4.33}$$

With this linear objective function and linear constraints one may expect an optimal solution to have $r^*_{0j} = 0$ for one or more $j \in \mathcal{J}$. As an example consider a case where for some $j' \in \mathcal{J}$ we have $c_{ij'} > c_{ij}$ for all $j \neq j'$ and all $i \in \mathcal{I}$, clearly the optimal solution will have $r_{0j'} = 0$.

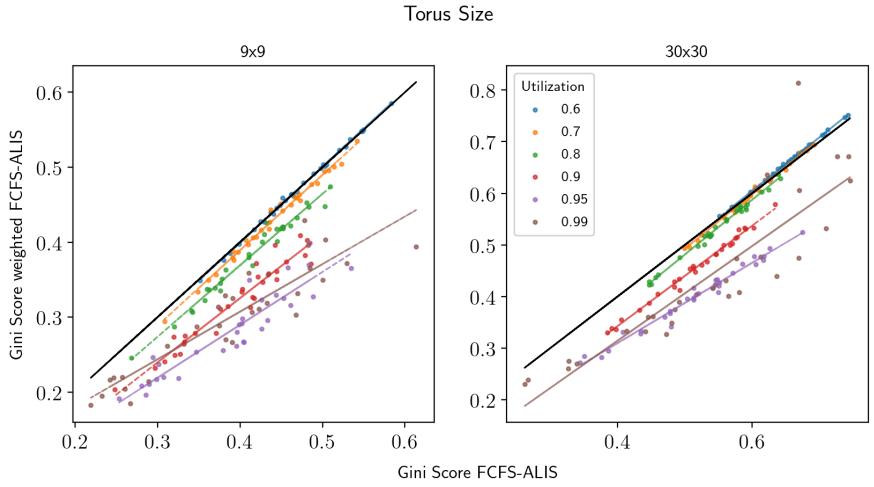


Figure 4.12: Gini Score in Torus graphs under weighted and standard polices

This will result in a system that maximizes the long term average reward by sending all customers to the same queue but, at the same time the queue lengths and waiting times will diverge. A way of avoiding such solutions is by adding the entropy function of (3.68) to the objective and instead solving the following convex optimization using the Sinkhorn-Knopp algorithm:

$$MaxUtil(G,\bar{\rho}): \quad \max_{r\in r_E,\in\mathbb{R}_n^+} (1-\gamma)\cdot\mathcal{Z}_c(\mathbf{r})+\gamma\cdot(\mathcal{H}(\mathbf{r})+\mathcal{H}_0(\mathbf{r})) \tag{4.34}$$
$$\text{subject to} \quad (3.71),\ (3.72)$$

for some $0<\gamma<1$. The gradient of this function is both negative and unbounded as $r_{ij}\to 0$ for any $(i,j)\in R$ and hence, given that the CRP holds and $\rho_{\mathcal{F}}<1$, the optimal solution must be an interior point of with $r^*_{ij}>0$ for all $(i,j)\in E$ and more importantly $r^*_{0j}>0$ for all $j\in\mathcal{J}$. Therefore, the use of the optimal solution $\mathbf{r}^*$ as assignment probabilities for an OQ system with Markovian routing will result in a stable system with finite long term average queue lengths. As descried in the previous section, equations (4.8),(4.9) and (4.30) can be applied to the set of rates $\mathbf{r}^*$ to produce weights for the set E. Our conjecture is that, when applied in a weighted-FCFS-ALIS or weighted-LQF-ALIS policy these weights will translate the trade-off of the avg. waiting time and long term avg. matching reward rate from the static optimization problem to the dynamic system. This trade-off could also be improved by instead maximizing the same reward objective but with the *min-max-fair* weighted entropy regularization term of (4.26) which was used to derive the weighed policies of Section 4.4 leading to the convex program:

$$MaxUtil(G,\bar{\rho}): \quad \max_{r\in r_E,\in\mathbb{R}_n^+} (1-\gamma)\cdot\mathcal{Z}_c(\mathbf{r})+\gamma\cdot(\mathcal{H}_{\bar{\rho}}(\mathbf{r})+\mathcal{H}_{\bar{\rho},0}(\mathbf{r})) \tag{4.35}$$
$$\text{subject to} \quad (3.71),\ (3.72)$$

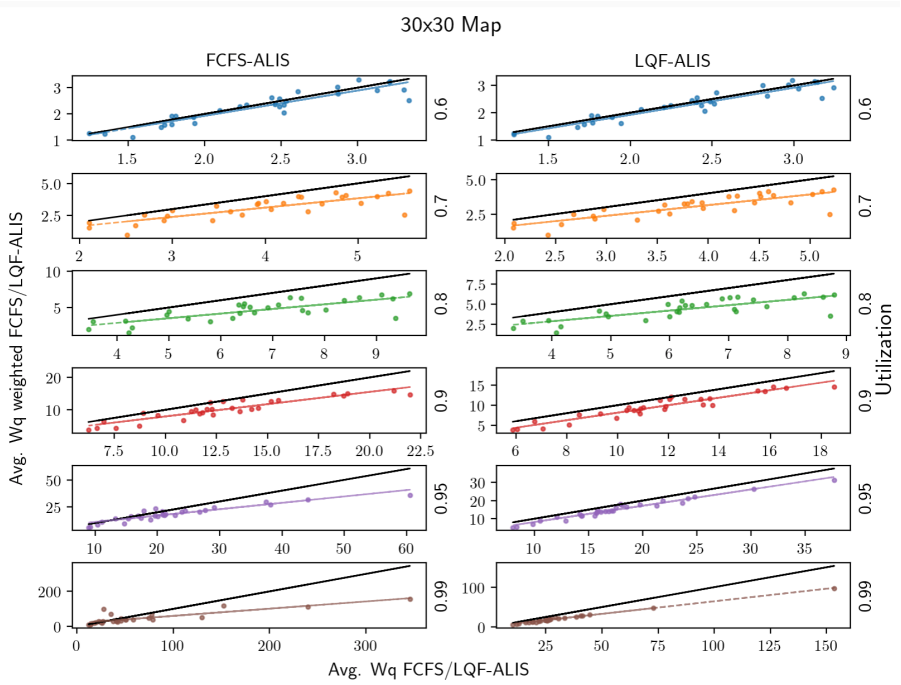


Figure 4.13: Avg. Wq in Map graphs under weighted and standard polices

As we will show in the subsequent section, the use of weights taken from the solution of (4.35) improves the reward rate to waiting time trade-off curve of the weighted polices compared to those obtained by the solution of (4.34). This improvement does come with a computational cost as adding the weights requires that the simple Sinkhorn-Knopp iteration that can be used to solve (4.34) be replaced with iterations of power sums that can quickly become numerically unstable. Hence we prefer to solve the optimization of (4.35) using a first order primal-dual mirror descent method. This is not a straight forward implementation as the norm of the gradient of the objective function, given by:

$$\nabla_{ij}\mathcal{H}_{\bar{\rho}}(\mathbf{r}) = w_{ij}(1 + \log(r_{ij})) \tag{4.36}$$

becomes unbounded as $r_{ij} \to 0$ for any $(i, j) \in E$ and the objective is therefore not a smooth function. The primal-dual method of [17] relies on the fact that $\mathcal{H}_{\bar{\rho}}(\mathbf{r})$ is 1-strongly convex and hence its dual function is smooth and thus by adding a norm-2 regularization term the regularized dual becomes a strongly convex, smooth, function and a mirror descant algorithm can be implemented to derive approximate optimal solutions for both the primal and dual problem with convergence guarantees.[1]. The standard uses of entropy terms such as (4.34),(4.35) in optimization are as barrier function and hence the term γ is usually set to be as small as possible with numerical considerations alone restricting the size of γ. In this

[1]we refer the reader to `https://github.com/dgrosbar/FSS` for details on the implantation

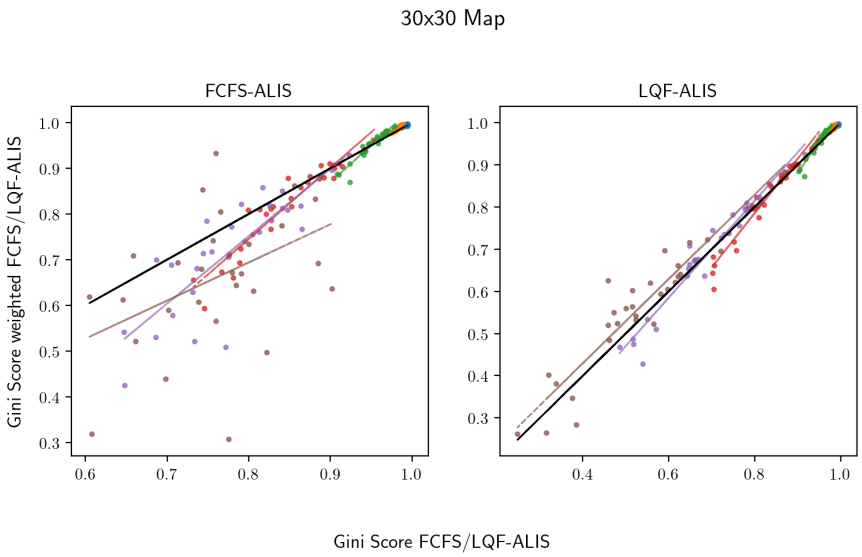


Figure 4.14: Gini score in Map graphs under weighted and standard polices

application the γ term is used to balance between a system that strives to reduce waiting times $\gamma \to 1$ and a system that strives to maximize the long term average rate of matching reward as $\gamma \to 0$. In order to use γ as a balancing factor we wish to normalize the two terms in the objective function so that both have similar contributions to the value of the objective function. In short, we wish to obtain the same optimal solution regardless of the units of c_{ij} and in order to do so we need to find a constant $\tilde{C}$ such that:

$$\tilde{C}\left(\overline{\mathcal{Z}}_c - \underline{\mathcal{Z}}_c\right) = \overline{\mathcal{H}} - \underline{\mathcal{H}} \tag{4.37}$$

where

$$\overline{\mathcal{Z}_c} = \max \mathcal{Z}_c(\mathbf{r}) \quad , s.t \quad (3.71),(3.72) \tag{4.38}$$
$$\underline{\mathcal{Z}_c} = \min \mathcal{Z}_c(\mathbf{r}) \quad , s.t \quad (3.71),(3.72) \tag{4.39}$$
$$\overline{\mathcal{H}} = \max \mathcal{H}(\mathbf{r}) + \mathcal{H}_0(\mathbf{r}) \quad , s.t \quad (3.71),(3.72) \tag{4.40}$$
$$\underline{\mathcal{H}} = \min \mathcal{H}(\mathbf{r}) + \mathcal{H}_0(\mathbf{r}) \quad , s.t \quad (3.71),(3.72) \tag{4.41}$$

In order to obtain $\overline{\mathcal{Z}_c}, \underline{\mathcal{Z}_c}$ we can either solve (4.31) or approximate it using a small $\gamma > 0$ and the Sinkhorn-Knopp projections. The value of $\overline{\mathcal{H}}$ can also be obtained using Sinkhorn-Knopp projections. The value of $\underline{\mathcal{H}}$ is not trivial to find however we use a lower bound given by:

$$\hat{\underline{\mathcal{H}}} = min\{\mathcal{H}(\boldsymbol{\lambda}), \mathcal{H}(\boldsymbol{\mu})\} \tag{4.42}$$

This is the lower bound as assigning each customer class to a single server class or vice versa is, if feasible, an entropy minimizing assignment. Having normalized the ranges of both

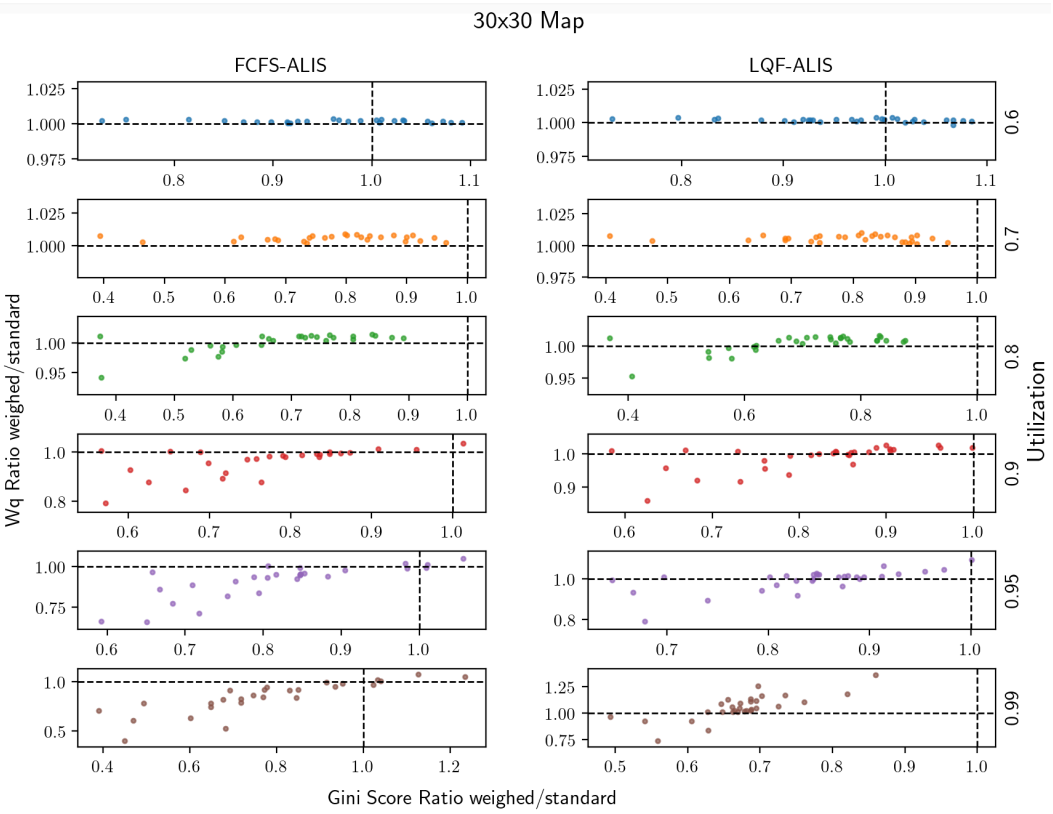


Figure 4.15: Ratio of Avg Wq against ratio of Gini score for Map systems

terms in the objective function the value of γ can now serve as a knob for a controller to trade-off between minimizing delays and maximizing the rate of reward collection.

4.7 Simulation Experiments - Reward Weighted Policies

Unlike the experiments of previous sections the output of a simulation experiment testing the performance of the weighted policies is a curve and not a point, as such we can not visually present the same quantity of results. Nonetheless we still present a rich set of results. First we take a broad sample of the experiments from the set of small graphs of Section 3.1. We simulate three traffic intensities with $\rho = .6, .8, .95$ and only consider the high and low density graphs. For each graph density we sample 5 graphs per utilization level and for each graph we sample 20(out of 40) pairs of arrival and service rate vectors. For each such set we randomly assign edge costs from a $Uniform[0, 10]$ distribution and simulate the system under both the weighted polices (4.35),(4.34) for $\gamma = .05, .1, .2, ..., .9, .95$ to generate the trade-off curves.

We do this with both the wFCFS-wALIS and wLQF-qALIS polices. For technical reasons we found it simpler to simulate a system with cost rather than one with rewards, the two are of course theoretically equivalent. The results of these simulations are presented in Figures 4.16 -4.19 where we plot the avg. waiting time of the system against the avg. rate of incurred cost, All figures show relatively similar results which lead to the following main conclusion

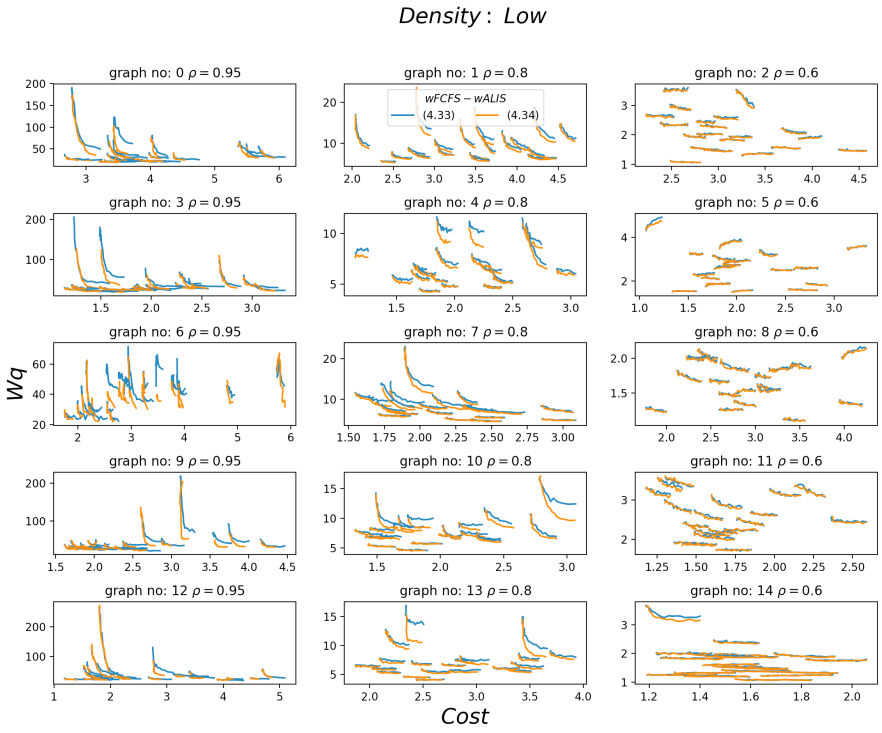


Figure 4.16: Wq-Cost curves for small low density Erdős-Rényi systems under wFCFS-wALIS

which is that the reward weighted wFCFS-wALIS, wLQF-wALIS with weights derived by (4.34),(4.35) are indeed a simple and effective tool for trading-off reward maximization and avg. waiting time reduction for these small scale systems. It is also apparent from the results that the weighted polices that use weights derived from the solution of (4.35) provide a more efficient trade-off curve than those based on weights derived from the optimal solution of (4.34). If the system is set to operate at a given avg. waiting time the policy weighted by (4.35) will produce lower avg. cost rate than the policy weighted by (4.34) and similarly, if the system is set to operate at given avg. cost rate, the policy weighted by (4.35) will produce lower avg. waiting time than the policy weighted by (4.34). However, this improved efficiency is not as predictable as we would like, for a fixed value γ we know that the (4.35) policy will either achieve a lower avg. waiting time or the avg. cost rate compared to the (4.34) yet we can not consistently predict which it will be. The results also indicate that,

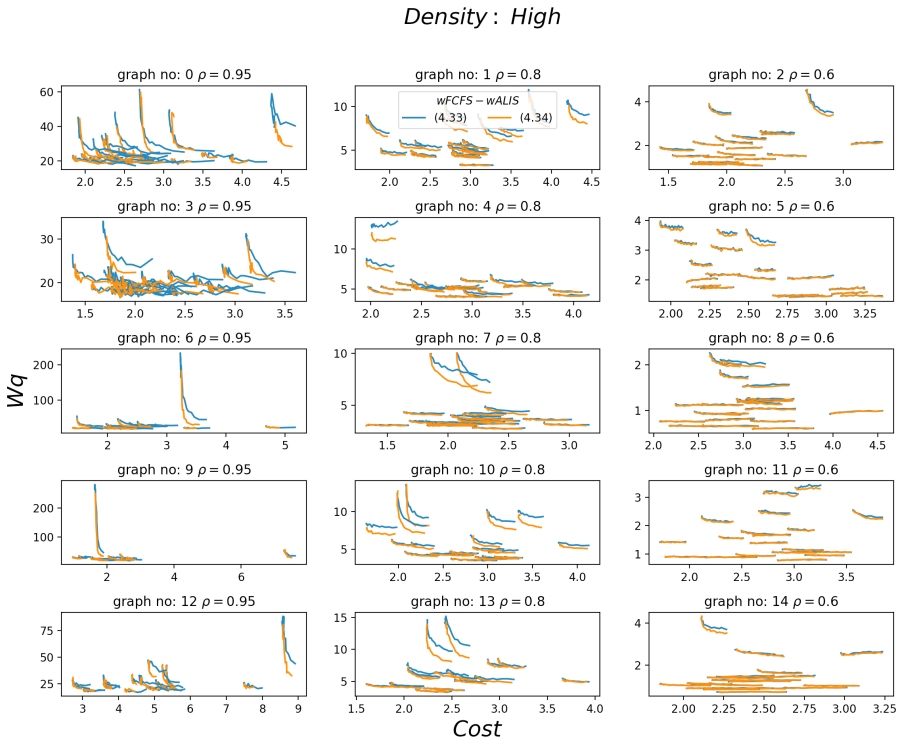


Figure 4.17: Wq-Cost curves for small high density Erdős-Rényi systems under wFCFS-wALIS

as one might expect, at lower traffic intensities a weighted policy can create a substantial reduction in the cost with only a minor increase of the avg. waiting time while at high traffic intensities the trade-off curve become much steeper and minor reductions in cost may cause a substantial increase of the avg. waiting time. For the larger scale graphs the computation required to replace each data point with a curve of values prevented us from simulating the entire set of instances, instead for we randomly choose 10 large instance of map systems and 10 large instance of tours systems. The Erdős-Rényi systems were not simulated as at that scale, as can be seen in the results of Section 4.5, the congestion is very low and the a reward maximizing policy is basically a greedy one. Out of the 10 instance of each type 5 systems were tested under the wFCFS-wALIS and 5 under the wLQF-wALIS policy. In each test we simulated the system under weighted policy for $\gamma = .1, .2, \ldots, .9$ once with the weights based on (4.34) and again with the weights based on (4.35).For the experiment we used the distance between the nodes as cost. Note that in the underlying construction the distance between to adjacent grid nodes is 5 and hence, since graph adjacency is defined by grid distance, for the 2-Torus instances the edge cots are $0, 5$ and 10 and for the Map instances they are $0, 5, 10$ and 15. We also simulate a Greedy policy that every decision assigns on the closet edge. Figures 4.20 - 4.23 plot the simulated curves. The first thing to observe is

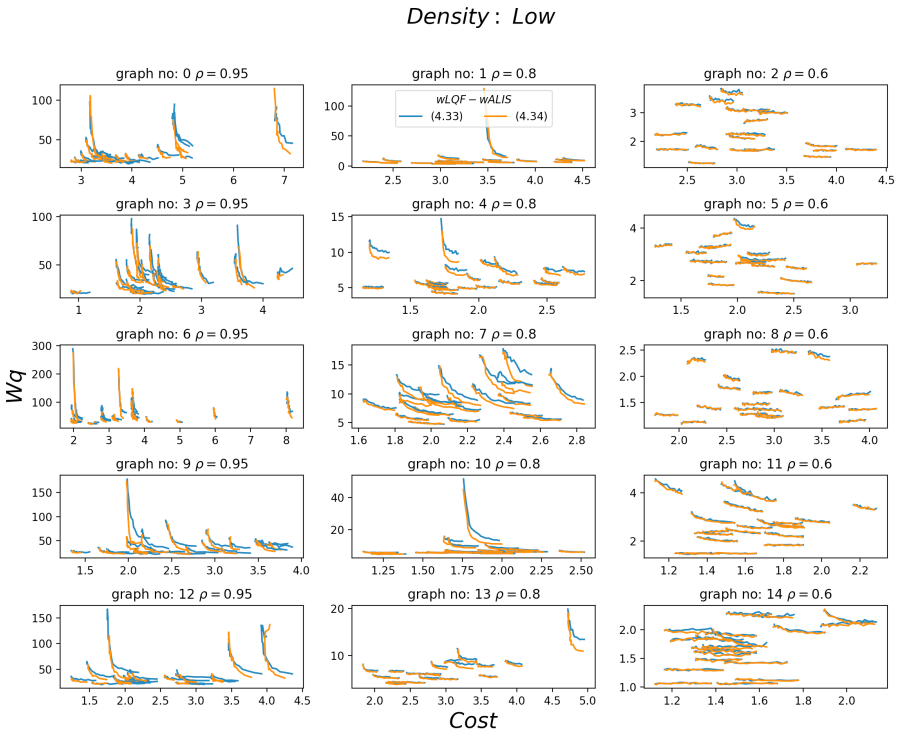


Figure 4.18: Wq-Cost curves for small low density Erdős-Rényi systems under wLQF-wALIS

that the curves are not as smooth as in the small cases and some have kinks for larger values of γ. Closer inspection revealed these are likely to be a result of numerical issues with the implementation of the primal-dual algorithm of [17]. Besides the kinks it appear that, in some cases, for higher values of γ the curve reverses direction and costs begin to increase the cost with γ. This as not as surprising as we can see that on some cases with traffic intensity of .6 the weighted policies achieve a lower cost than a lower cost than the a Greedy policy. This suggests that becoming too greedy at some point becomes counter productive for the purpose of reducing costs. Furthermore, the weighted policies are based on the optimization problems in (4.34),(4.35) and account for the arrival rates while a greedy policy is has no notion of future expected arrivals. Finally, we see that, in some case, for these larger systems the policies based on (4.34) provided more efficient trade-offs for high values of γ. This is at least in part due to the additional numerical complexity associated with solving (4.35) over (4.34) which increases considerably as γ approach's unity. Nonetheless, in every experiment conducted the policies based on (4.35) provided a more efficient trade-off for a wide range of operating points, especially for those points which provide reasonable waiting times.

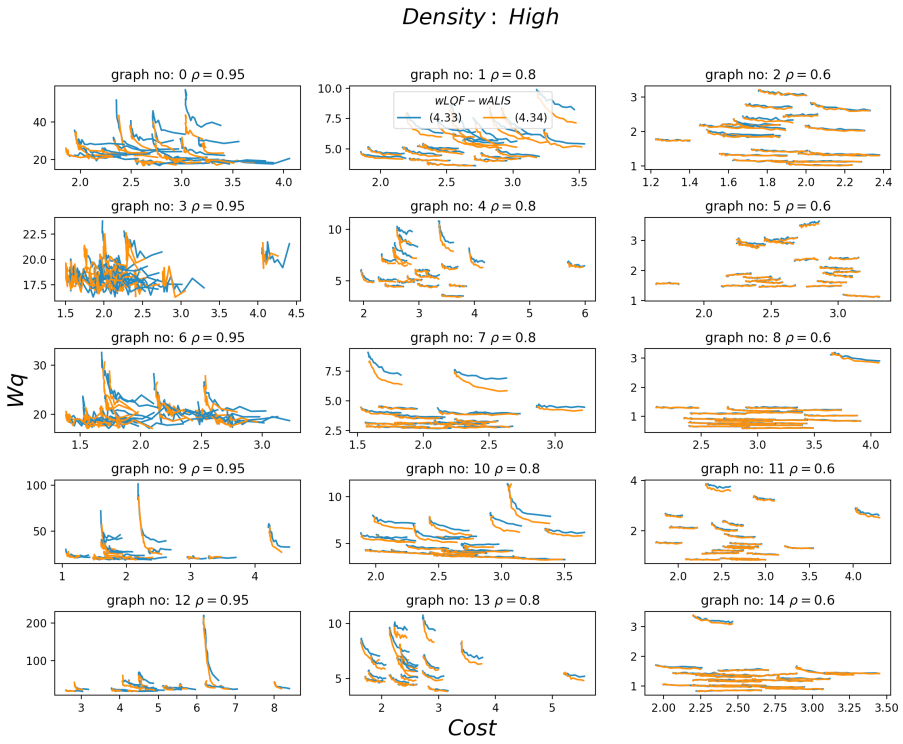


Figure 4.19: Wq-Cost curves for small high density Erdős-Rényi systems under wLQF-wALIS

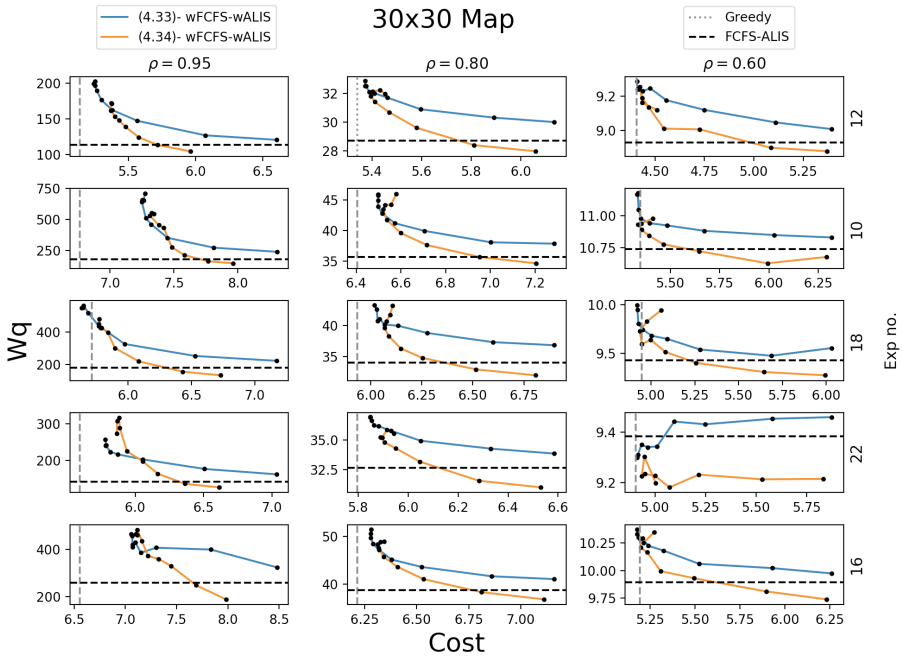


Figure 4.20: Wq-Cost curves for Map systems under wFCFS-wALIS policies

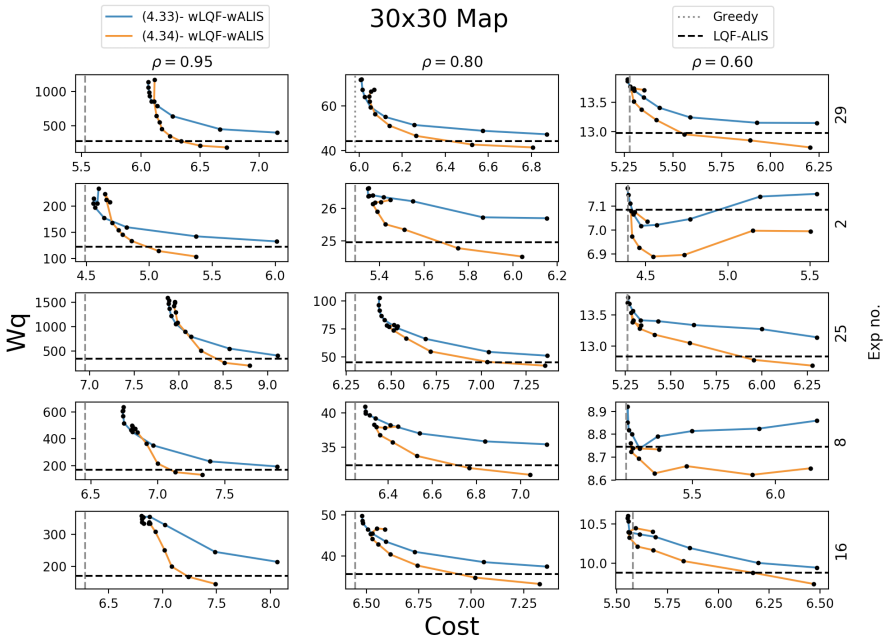


Figure 4.21: Wq-Cost curves for Map systems under wLQF-wALIS policies

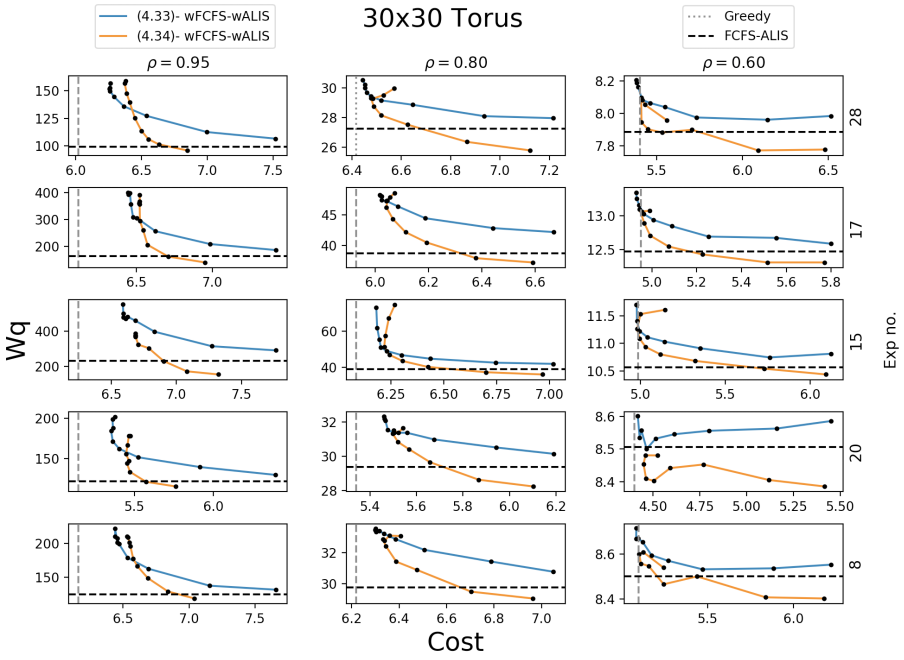


Figure 4.22: Wq-Cost curves for Torus systems under wFCFS-wALIS policies

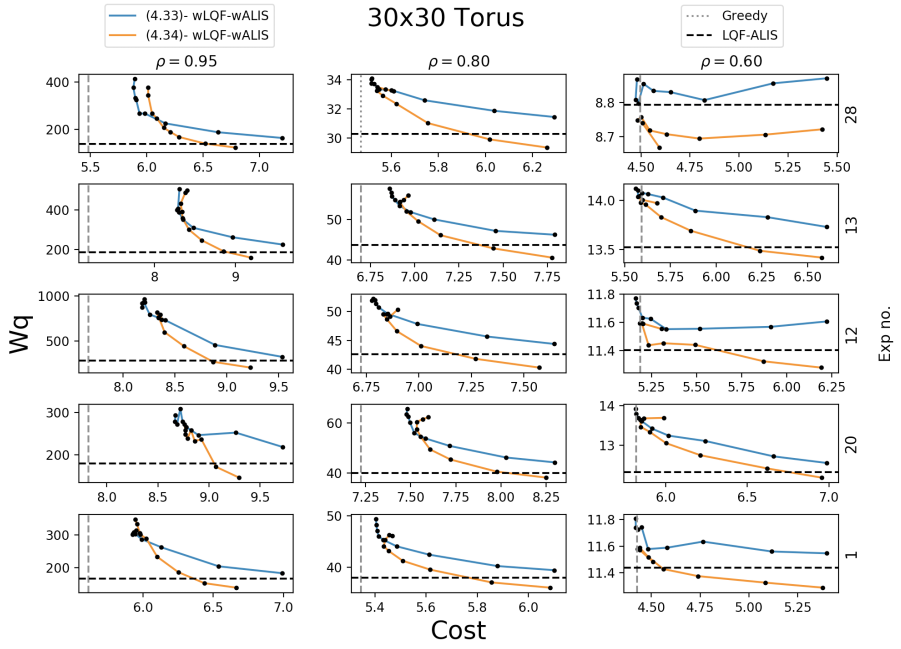


Figure 4.23: Wq-Cost curves for Torus systems under wLQF-wALIS policies

Chapter 5

Summary and Future Research

To conclude we first wish to summarize the main contributions of this work. Chapter 1 provides a formulation of the SBPSS model and a review of related literature. In this Chapter we also motivate the subsequent work by arguing that control and approximation of sub-critical SBPSS has been relatively neglected in literature despite the prevalence of such system in applications. In Chapter 2 we define the min-max-fair decomposition of a SBPSS and discussed related structural properties of the decomposition. In the later part of section 2.1 we described a procedure that uses a parametric minimum cut algorithm to obtaining the min-max-fair decomposition of a the SBPSS. The first main contribution of this work appears in Chapter 3 that is concerned with approximations of the matching rates of a SBPSS under the FCFS-ALIS and LQF-ALIS policies. First, in section 3.1 we derived the approximation of [14] as a maximum entropy based approximation and demonstrated by simulation experiment the advantage it has over the approximations of [20] and [4] achieving an absolute error ratio of 3% across all test cases. In section 3.2 we defined the infinite ALIS bipartite matching sequence. We then went on to formulate a novel, fluid dynamic based scheme for approximating the matching rates of the sequence and presented experiment results showing the approximation predicts the ALIS matching rates with an absolute error ratio of 3%-5% for both small and large scale systems. A key practical contribution in the paper appeared in section 3.3 where, by adjusted and combining both the ALIS and FCFS matching sequences we provide an approximation scheme for the matching rates of a sub critical SBPSS under both the FCFS-ALIS and LQF-ALIS policies, achieving error rates of 3% on small scale graphs and 5%-7% for large scale graphs for either low ($< .1$) or high ($> .75$) traffic intensities. This to the best of our knowledge is the first approximation of this kind. In Chapter 4 we attempt to leverage upon the approximations and related insights of Chapter 3 to improve upon the standard FCFS/LQF-ALIS policies. In order to do so in section 4.3 we observed the relation between the approximate matching rates of an SBPSS under the FCFS/LQF-ALIS policies and the queue length distribution of an analogous output-queued system. Through the output-queued system analogy we exposed the inherent flaws of these commonly used policies. Based on the approximations of Chapter 3 and the output-queued analog in section 4.4 we formulate the min-max-fair weighted max

entropy convex program. The optimal solution of the program is then used to construct weighted FCFS-ALIS and LQF-ALIS polices for the dynamic control of an SBPSS. Results of extensive simulation experiments presented in section 4.5 show these weighted polices provide a substantial performance improvement over the standard policies reducing both the avg. waiting times in the system and the variance of avg. waiting times across customer classes, especially for those instances where the standard polices incurred long avg. waiting times. Finally, in section 4.6 we extend the weighted scheme to accommodate systems with matching rewards and allow a controller to systematically trade-off between the avg. waiting time and long term avg. reward rate. Two methods for deriving the weights are presented, the first by solving an optimal transport problem with standard entropic regularization (4.34) and the second by solving the same optimal transport problem with the min-max-fair entropic regularization (4.35). The trade of curve is explored in section 4.7 through large scale simulations. The results show the weighted policies using weights derived by solving (4.35) provide a trade-off curve that dominates the curve obtained by the polices with weights obtained by solving(4.34). At this points we wish to clearly state two theoretical and practical gaps in the work. First, the fixed point iteration ALIS approximation does not have any known convergence guarantees and furthermore, our experiments have shown the algorithm does not converge for instances where the CRP condition does not hold and hence the properties of the algorithm should be further explored. Second, although the experiments of Sections 4.5, 4.7 clearly demonstrate the trade-off between avg. waiting tine and avg. reward rate we do not yet have any analytical method to estimate the curve and hence application of the weighted polices for this purpose requires either simulation or real time tuning of the γ parameter to achieve the desired operating point. The work in this paper suggests multiple interesting paths for further research in three main branch; approximations, control policies and system design. The first branch regards further approximations for SBPSSs and other related matching systems. To this extent, an immediate research work would be closing the gap in the approximation scheme for systems without CRP. We conjecture this could be done by a two stage scheme in which the min-max-fair subsystems are first approximated in isolation and then, based on the isolated approximations, the system is approximated as a whole. Another obvious approximation of interest is that of the matching rates obtained by the weighted polices in Chapter 4. Initial investigations indicate that, despite having at times a significant impact on the waiting time distribution, the weighted policies have a much smaller impact on the matching rates and hence it may be the case that those too can be well approximated by a variant of (3.71),(3.72). The approximation schemes of 3.3 can also be extended to non-bipartite matching systems with abandonment which have been recently used to model passenger route matching in ride-sharing [41], [8]. Having established the accuracy of the matching rate approximations in section 3.3 one can use the matching rate approximations to derive waiting time approximations for the SBPSS. This seems a reasonable goal as methods for obtaining the Laplace-Stieltjes transform of the waiting time from the matching rates have been described in[53] and may be coupled with numerical methods for the inverse transform to derive approximations. In this work we restricted the scope only to non-idling policies, in practice most physical systems operate

under some batching window. The batching window may be a long one in which many customers arrive and many servers become available. In such case it may be appropriate use of optimal matching algorithms such as the one presented in [49], if the batching window is long enough the expected matching rate will converge to the optimal transport solution problem of 4.31. However, in many practical applications the batching window is small and, if we consider a non-idling policy to be an extreme case of a zero length batching window, we may expect that similar entropy regularized optimal transport weights can be applied to improve the matching algorithm. In a broader sense, as illustrated in Figure 5.1 we are interested in exploring how, through entropic regularization, the optimal transport solution may inform matching policies on much shorter batching windows. The final branch for future

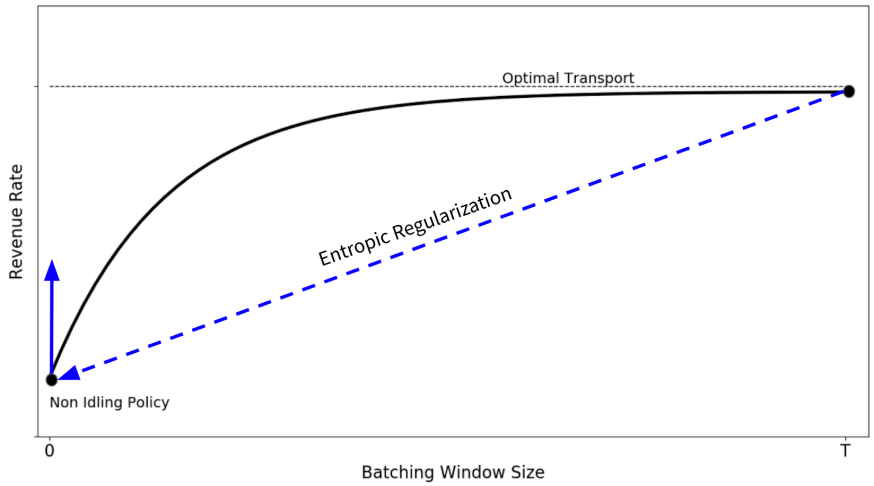


Figure 5.1: Optimal transport with entropic regularization applied to policies with a short batching windows

research is optimal system design. In a recent paper ([4]) the authors use the quadratic approximation of (3.50), (3.49) to find an approximated optimal compatibility matrix for an SBPSS with predefined arrival rates and identical service rates under heavy traffic. If the approximation of (3.71),(3.72), which has bee shown in Section 3.1 to be more accurate than the QP approximation used in [4], were to be used instead in the approximation scheme it would enable the optimal design of a much broader range of systems with sub critical workload and homogeneous service.

Bibliography

[1] Ivo Adan, Rhonda Righter, and Gideon Weiss. "FCFS parallel service systems and matching models". In: *Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools*. ACM. 2017, pp. 106–112.

[2] Ivo Adan and Gideon Weiss. "A skill based parallel service system under FCFS-ALIS—steady state, overloads, and abandonments". In: *Stochastic Systems* 4.1 (2014), pp. 250–299.

[3] Ivo Adan and Gideon Weiss. "Exact FCFS matching rates for two infinite multitype sequences". In: *Operations research* 60.2 (2012), pp. 475–489.

[4] Philipp Afeche, Rene Caldentey, and Varun Gupta. "On the Optimal Design of a Bipartite Matching Queueing System". In: *Available at SSRN 3345302* (2019).

[5] Mor Armony and Amy R Ward. "Fair dynamic routing in large-scale heterogeneous-server systems". In: *Operations Research* 58.3 (2010), pp. 624–637.

[6] Mor Armony et al. "On patient flow in hospitals: A data-based queueing-science perspective". In: *Stochastic Systems* 5.1 (2015), pp. 146–194.

[7] Siddhartha Banerjee, Ramesh Johari, and Carlos Riquelme. "Pricing in ride-sharing platforms: A queueing-theoretic approach". In: *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. ACM. 2015, pp. 639–639.

[8] Siddhartha Banerjee, Yash Kanoria, and Pengyu Qian. "State dependent control of closed queueing networks with application to ride-hailing". In: *arXiv preprint arXiv:1803.04959* (2018).

[9] Amir Beck and Marc Teboulle. "Mirror descent and nonlinear projected subgradient methods for convex optimization". In: *Operations Research Letters* 31.3 (2003), pp. 167–175.

[10] Dimitri P Bertsekas. "Nonlinear programming". In: *Journal of the Operational Research Society* 48.3 (1997), pp. 334–334.

[11] Lev M Bregman. "Proof of the convergence of Sheleikhovskii's method for a problem with transportation constraints". In: *USSR Computational Mathematics and Mathematical Physics* 7.1 (1967), pp. 191–204.

[12] Richard A Brualdi, Herbert John Ryser, et al. *Combinatorial matrix theory*. Vol. 39. Springer, 1991.

[13] René A Caldentey and Edward H Kaplan. "A heavy traffic approximation for queues with restricted customer-server matchings". In: (2007).

[14] René Caldentey, Edward H Kaplan, and Gideon Weiss. "FCFS infinite bipartite matching of servers and customers". In: *Advances in Applied Probability* 41.3 (2009), pp. 695–730.

[15] Lidia Ceriani and Paolo Verme. "The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini". In: *The Journal of Economic Inequality* 10.3 (2012), pp. 421–443.

[16] Bala G Chandran, Dorit S Hochbaum, and Operations Research. "A Computational Study of the Pseudoflow and Push-Relabel Algorithms for the Maximum Flow Problem". In: *Operations Research* 57.2 (2009), pp. 358–376.

[17] Alexey Chernov, Pavel Dvurechensky, and Alexander Gasnikov. "Fast primal-dual gradient method for strongly convex minimization problems with linear constraints". In: *International Conference on Discrete Optimization and Operations Research*. Springer. 2016, pp. 391–403.

[18] S H Chung, Chi-Yo Huang, and A H I Lee. "Capacity allocation model for photolithography workstation with the constraints of process window and machine dedication". In: *Production Planning & Control* 17.7 (2006), pp. 678–688.

[19] Marco Cuturi. "Sinkhorn distances: Lightspeed computation of optimal transport". In: *Advances in neural information processing systems*. 2013, pp. 2292–2300.

[20] Mohammad M Fazel-Zarandi and Edward H Kaplan. "Approximating the First-Come, First-Served Stochastic Matching Model with Ohm's Law". In: *Operations Research* 66.5 (2018), pp. 1423–1432.

[21] Barak Fishbain, Dorit S Hochbaum, and Stefan Mueller. "A competitive study of the pseudoflow algorithm for the minimum s–t cut problem in vision applications". In: *Journal of Real-Time Image Processing* 11.3 (2016), pp. 589–609.

[22] Alan Frieze and Michał Karoński. *Introduction to random graphs*. Cambridge University Press, 2016.

[23] Giorgio Gallo, Michael D Grigoriadis, and Robert E Tarjan. "A fast parametric maximum flow algorithm and applications". In: *SIAM Journal on Computing* 18.1 (1989), pp. 30–55.

[24] Kristen Gardner et al. "Queueing with redundant requests: exact analysis". In: *Queueing Systems* 83.3-4 (2016), pp. 227–259.

[25] Silviu Guiasu. "Maximum entropy condition in queueing theory". In: *Journal of the Operational Research Society* 37.3 (1986), pp. 293–301.

[26] Itay Gurvich, Mor Armony, and Avishai Mandelbaum. "Service-Level Differentiation in Call Centers with Fully Flexible Servers". In: *Management Science* 54.2 (2008), pp. 279–294.

[27] Itay Gurvich and Ward Whitt. "Queue-and-Idleness-Ratio Controls in". In: *Many-Server Service Systems. Mathematics of Operations Research* 34.2 (2009), pp. 363–396.

[28] J Michael Harrison and Marcel J López. *Heavy traffic resource pooling in parallel-server systems*. Tech. rep. 1999, pp. 339–368.

[29] Dorit S Hochbaum and Operations Research. "The Pseudoflow Algorithm: A New Algorithm for the Maximum-Flow Problem". In: *Operations Research* 56.4 (2008).

[30] Shlomo Hoory, Nathan Linial, and Avi Wigderson. "Expander graphs and their applications". In: *Bulletin of the American Mathematical Society* 43.4 (2006), pp. 439–561.

[31] Raj Jain, Dah-Ming Chiu, and William R Hawe. *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Vol. 38. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984.

[32] Selmer M Johnson. "Generation of permutations by adjacent transposition". In: *Mathematics of computation* 17.83 (1963), pp. 282–285.

[33] Carl Johnzen et al. "Importance of qualification management for wafer fabs". In: *Advanced Semiconductor Manufacturing Conference, 2007. ASMC 2007. IEEE/SEMI*. IEEE. 2007, pp. 166–169.

[34] Edward H Kaplan. "Managing the demand for public housing". PhD thesis. Massachusetts Institute of Technology, 1984.

[35] Hongseok Kim et al. "Distributed alpha-optimal user association and cell load balancing in wireless networks". In: *IEEE/ACM Transactions on Networking* 20.1 (2012), pp. 177–190.

[36] Philip A Knight. "The Sinkhorn–Knopp algorithm: convergence and applications". In: *SIAM Journal on Matrix Analysis and Applications* 30.1 (2008), pp. 261–275.

[37] Demetres D Kouvatsos. "Entropy maximisation and queueing network models". In: *Annals of Operations Research* 48.1 (1994), pp. 63–126.

[38] Yi Lu et al. "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services". In: *Performance Evaluation* 68.11 (2011), pp. 1056–1071.

[39] Avishai Mandelbaum and Alexander L Stolyar. "Scheduling Flexible Servers with Convex Delay Costs: Heavy-Traffic Optimality of the Generalized cμ-Rule". In: *Operations Research* 52.6 (2004).

[40] Sean Meyn. "Stability and asymptotic optimality of generalized MaxWeight policies". In: *SIAM Journal on Control and Optimization* 47.6 (2009), pp. 3259–3294.

[41] Mohammadreza Nazari and Alexander L Stolyar. "Reward maximization in general dynamic matching systems". In: *Queueing Systems* 91.1-2 (2019), pp. 143–170.

[42] Hirotaka Sakasegawa. "An approximation formula L q= $\alpha\cdot$ ρ β/(1-ρ)". In: *Annals of the Institute of Statistical Mathematics* 29.1 (1977), pp. 67–75.

[43] Richard Sinkhorn and Paul Knopp. "Concerning nonnegative matrices and doubly stochastic matrices". In: *Pacific Journal of Mathematics* 21.2 (1967), pp. 343–348.

[44] Alexander L Stolyar. "Maximizing Queueing Network Utility Subject to Stability: Greedy Primal-Dual Algorithm". In: *Queueing Systems* 50 (2005), pp. 401–457.

[45] Alexander L Stolyar and Yuan Zhong. "A greedy randomized algorithm achieving sublinear optimality gap in a dynamic packing model". In: *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*. IEEE. 2016, pp. 319–326.

[46] Alexander L Stolyar and Yuan Zhong. "Asymptotic optimality of a greedy randomized algorithm in a large-scale service system with general packing constraints". In: *Queueing Systems* 79.2 (2015), pp. 117–143.

[47] Alexander L Stolyar et al. "Control of systems with flexible multi-server pools: a shadow routing approach". In: *Queueing Systems* 66.1 (2010), pp. 1–51.

[48] Rishi Talreja and Ward Whitt. "Fluid Models for Overloaded Multiclass Many-Server Queueing Systems with First-Come, First-Served Routing". In: *Management Science* 54.8 (2008), p. 1513.

[49] John N Tsitsiklis and Kuang Xu. "Flexible Queueing Architectures". In: *Operations Research* 65.5 (2017), pp. 1398–1413.

[50] John N Tsitsiklis, Kuang Xu, et al. "On the power of (even a little) resource pooling". In: *Stochastic Systems* 2.1 (2012), pp. 1–66.

[51] C P L Veeger, L F P Etman, and J E Rooda. "Generating cycle time-throughput-product mix surfaces using effective process time based aggregate modeling". In: *Proceedings of 13th ASIM conference*. 2008, pp. 519–529.

[52] Hector Velayos, Victor Aleo, and Gunnar Karlsson. "Load balancing in overlapping wireless LAN cells". In: *Communications, 2004 IEEE International Conference on*. Vol. 7. IEEE. 2004, pp. 3833–3836.

[53] Jeremy Visschers, Ivo Adan, and Gideon Weiss. "A product form solution to a system with multi-type jobs and multi-type servers". In: *Queueing Systems* 70.3 (2012), pp. 269–298.

[54] Rodney B Wallace and Ward Whitt. "A staffing algorithm for call centers with skill-based routing". In: *Manufacturing & Service Operations Management* 7.4 (2005), pp. 276–294.

[55] Ruth J Williams. "Diffusion approximations for open multiclass queueing networks: sufficient conditions involving state space collapse". In: *Queueing systems* 30.1-2 (1998), pp. 27–88.

[56] A G Wilson. *A Statistical Theory of Spatial Distribution Models*. Tech. rep. 1967, pp. 253–269.

[57] Alan Geoffrey Wilson. "The use of the concept of entropy in system modelling". In: *Journal of the Operational Research Society* 21.2 (1970), pp. 247–265.